University of Alberta

Equipercentile Methods of Linking Inter-regional and Regional Assessments

by

Fernando Cartwright

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the

requirements for the degree of Master of Education

Department of Educational Psychology

Edmonton, Alberta

Spring 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-82225-7

Canada

University of Alberta

Library Release Form

Name of Author: Fernando Cartwright

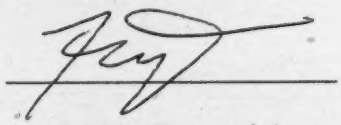
Title of Thesis: Equipercentile Methods of Linking Inter-regional and Regional Assessments

Degree: Master of Education

Year this Degree Granted: 2003

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.



#807, 110 - Forward Avenue
Ottawa, Ontario K1Y 4S9
CANADA

April 4, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Equipercentile Methods of Linking Inter-regional and Regional Assessments* submitted by *Fernando Cartwright* in partial fulfillment of the requirements for the degree of Master of Education.

Todd Rogers

Mark Gierl

Ernest Skakun

April 4, 2003

ABSTRACT

Large-scale assessments are typically administered by regional institutions. However, regional student populations are often administered assessments that span many regions (e.g., national or international assessments). Although the independent assessments may be similar in content, comparison of results requires a formal link between their different scales. Using international and regional assessment data from British Columbia, this study examines methods of continuising non-normal score distributions for the application of equipercentile scale linking. The study develops algorithms for the implementation of these methods and the incorporation of measurement error, linking error, and sampling error in reported group statistics. Methods developed in this study may be generalised to other regional and inter-regional assessment linkages using a variety of scaling methods.

ACKNOWLEDGEMENT

This work would not have been possible without the support of many people at many stages of its conception and my academic pursuits. I would like to thank Xin Ma for the invitiation to study the field of measurement; my life would undoubtedly have followed a much different path had I not accepted it. Everyone from CRAME, particulary Todd Rogers and Mark Gierl, helped to provide explanations, guidance, perspective, and humour, provided when needed.

I would also like to thank many people who provided invaluable assistance outside of my formal program. The most unexpectedly pleasant experience of both my graduate studies and composition of this thesis has been the acquaintance of Steve Hunka, who, perhaps inadvertently, has helped me learn more about mathematics and programming than I had ever expected of myself. Thanks to my colleagues at Statistics Canada and the British Columbia Ministry of Education, without whose inspiration and cooperation I might never have completed my program. Above all, my greatest appreciation and gratitude go to my wife, Holly, for her plentiful supply of support and patience. I am glad that things turned out well.

TABLE OF CONTENTS

INTRODUCTION ..... **ERROR! BOOKMARK NOT DEFINED.**

LINKING SCALES OF DIFFERENT ASSESSMENT 3

Types of Scale linkages *3*

Equating 4

Calibration 4

Projection 4

Moderation 5

Linkages between regional and inter-regional assessment *5*

Linear Scale Linking Methods 6

Equipercentile Linking Methods 8

PURPOSE OF THE STUDY 12

METHOD 13

DATA 13

Instruments *13*

Content Equity *15*

Sample *17*

Properties of Score Distributions *17*

Equity in Measurement Precision *18*

PROCEDURE 19

Necessary Assumptions for Estimating the Linking Function *19*

Estimation of the Equipercentile Linking Function *21*

Influence of Measurement Error *24*

Step 1: Estimating Cumulative Density Functions *26*

Gaussian Kernel Density Estimation 27

Finite mixture models 28

Variable component mixture models 29

Fixed Mixture Models 30

4-Parameter Beta Distributions 31
Step 2: Estimation of Inverse Cumulative Density Functions *32*
Step 3: Estimation of Score Equivalents *33*
Computational Considerations *34*
EVALUATION OF DENSITY ESTIMATORS 35
UNCERTAINTY IN TRANSFORMED SCORES: MEASUREMENT ERROR, LINKING ERROR, AND SAMPLING ERROR 37
Error in Scores for Individuals *37*
Linking Error 38
Measurement Error 39
Error in Estimated Sample Statistics *40*
Measurement and Linking Variance 41
Combined Error for Means 41
Combined Error for Proportions 42
Sampling Variance 42
EVALUATION OF LINKING PROCEDURES 43
COMPUTER PROGRAMS 44
RESULTS 44
DISCUSSION 50
LIMITATIONS OF THE STUDY 52
FUTURE RESEARCH 55
Mixture Models *55*
Linking Items Using Score-Based Equipercentile Linking Functions *56*
Validity of Inter-regional Linkages *57*
REFERENCES 59
APPENDIX 64

LIST OF TABLES

Table 1 Comparison of Item Content Representation in FSA and PISA as a Percentage of Total Items 16
Table 2 Comparison of Empirical Central Moments to Fitted Central Moments across Estimation Techniques 45
Table 3 Distributional Characteristics of Rescaled Scores 47
Table 4 Standard Errors of Rescaling Corresponding to Performance Level Cut-points 49
Table 5 Results of Applying Linking Functions to Transform Cut-points 50

LIST OF FIGURES

Figure 1. Comparison of continuous density estimation methods 46
Figure 2. Comparison of standard error of linking functions for FSA-to-PISA and PISA-to-FSA 48
Figure 3. Invariance of the FSA-to-PISA linking function 53
Figure 4. Invariance of the FSA-to-PISA linking function 54

Equipercentile Methods of Linking Inter-regional and Regional Assessments

Often, separate provincial, federal, and international government and agencies, such as provincial education ministries and departments; the Council of Ministers of Education, Canada (CMEC), and the Organization for Economic Cooperation and Development (OECD) create and administer tests independently of each other. Some of these large-scale assessments are intended for use only within a region (e.g. a specific province or country), while others are intended to provide information across regions. However, the majority of these independent assessments tend to assess the same general domains of proficiency, such as reading, writing, or numeracy skills. For example, both the Program for International Student Assessment (PISA), initiated by the OECD in 32 countries, and the School Achievement Indicators Programme (SAIP), administered by the CMEC in all Canadian educational jurisdictions, have schedules for periodic assessment of reading, mathematics, and science (*PISA 2000 Technical Report*, 2002; *SAIP 2001 Technical Report*, in press). These content domains are also assessed by provincial testing programs in many educational jurisdictions in Canada.

The scores associated with specific large-scale assessments tend to have known distributional properties, such as means, standard deviations, percentiles and shapes (skewness and kurtosis). As a result, the scales develop intrinsic meaning and are widely interpretable. Unfortunately, while what is assessed is similar – if not the same – across assessments, the scores are reported on different scales. For example, provincial assessments in Alberta are reported on a scale that represents performance on a universe of relevant items as a probability of achieving a desired outcome. An observed score of 50 out of a possible 100 is generalized to represent the expectation that any examinee achieving that score has a 50% chance of correctly responding to a randomly chosen item in the test domain. Furthermore, this generalization can be interpreted against standards

previously set for performance in the domain. For example, a 50% probability of success is often considered an acceptable level of performance, and any examinee with a score above 50% is considered to have "passed" the standard. SAIP results, on the other hand, are expressed as categorical scores on an ordinal scale, which is defined by described performance levels. Many other assessments, such as PISA, use Item Response Theory (IRT) to scale results and report scores on an arbitrary scale that represents the latent trait (e.g., reading ability). While each of these methods is designed to meet an intended purpose, their dissimilarity makes interpretation across assessments impossible.

Nevertheless, regional institutions may wish to know how the properties of their scales translate to the scales of inter-regional assessments. For example, how does a score of 70% on a provincial reading test translate to a performance level or latent trait score on reading test used to compare performance of several provinces or countries? This question may take the form of national educational institutions wishing to compare standards of performance against an international assessment or sub-national institutions comparing against national or international assessments. This approach has been somewhat formalised in the United States, where the National Council on Educational Standards and Testing (1992), stressed the importance of linking regional educational achievement to national standards. Specifically, many states estimate the achievement of their students on the National Assessment of Educational Progress (NAEP) based on their state assessment results. In Canada, despite the presence of a national assessment, the political division of responsibility for education has precluded the idea of national standards. However, as shown by the ongoing SAIP and participation in PISA, there is a growing interest at the provincial level in the inter-regional comparison of educational outcomes. By revealing patterns of performance on a common metric, linking assessments allows comparison of performance across different educational systems and standards. This, in turn, may enable researchers and policy makers to compare the effectiveness of different educational policies and

practices and promotes fairness in educational opportunity across systems through the standardization of educational expectations.

As described by Mislevy (1992) and Linn (1993), the problem of linking assessments is one of inference. For any assessment, X, which is used to make inferences about ability in a specific domain, there is a desire to determine the extent to which scores on another assessment, Y, can be used to make those same inferences. There will be an association, or link, between the scores of X and Y for any group of students. The goal is to determine the nature of the association between the evidence that two measures give about the domain of interest and making valid interpretations. Previously, the term *equity* in linking assessments has been used to describe the psychometric equivalence of the tests to be linked in terms of expected scores and standard errors of measurement (Lord, 1980; Morris, 1982; Yen, 1983). However, this term is used under the assumption that test forms to be linked are based on the same test specifications and therefore have identical content representation (Kolen & Brennan, 1995, Chapter 1). As used below, the term *equity* will be used more broadly to describe how similar the assessments to be linked are in terms of their goals, content coverage, as well as measurement properties. The validity of interpretations is limited by how well the two assessments meet the assumption of equity. The greater this assumption is met, the stronger the linking relationship will be.

LINKING SCALES OF DIFFERENT ASSESSMENT

Types of Scale linkages

Mislevy (1992) and Linn (1993) define four types of linkages between scales, ordered from strongest to weakest according to how well the linked assessments meet the assumption of equity: equating (strongest), calibration, projection, and moderation (weakest). The characteristics and limitations of each type are described in greater detail below.

Equating

Equating is justified if the assessments to be linked are built to the same test specifications, and are administered to the same population. However, after administration, the two assessments are found to have slightly different scale properties, and a linking procedure is necessary in order to use common scale reference points. In order for scales of assessments to be considered equated, they must have complete matches in content coverage, difficulty, type of questions used, mode of administration, test length, and measurement precision. When scores are equated, any appropriate use or inference of scores on assessment Y can be also be made using the score equivalent from assessment X.

Calibration

Calibration occurs when the two assessments to be linked are measuring the same construct, but have been designed to have different measurement precision or level of difficulty. In calibrating two assessments, the goal is to adjust their scales so that the expected value of an examinee's score across a large number of observations is the same across both tests. For any single pair of observations, however, the linked scores from the two assessments might be different, because one assessment would have greater measurement error.

Projection

Projection is used to predict (or project) how examinees might perform on an assessment based on results on another assessment built to a different set of specifications. For example, two tests assessing the same broad skill domain may have unequal representations of the tasks within that domain. Unlike calibration and equating, the relationship between assessment X and assessment Y is not assumed to be constant across groups or over time. As equity between the two assessments decreases, the linking function is increasingly influenced by spurious

sources of information. These sources may include changes to curriculum or focus of instruction.

Moderation

Moderation provides the weakest link between assessments. It occurs when the goals of the assessments are different. For example, the assessments may be designed to measure different constructs or to assess different populations. There are two types of moderation: statistical moderation and social moderation. Statistical moderation uses the mathematical machinery of equating, but recognises that the assessments are measuring different constructs. Social moderation, on the other hand, involves making judgements about the normative or relative importance of performance on different assessments.

Linkages between regional and inter-regional assessment

Estimating a linking function between regional and inter-regional assessments falls somewhere between projection and moderation, and therefore provides weak results in terms of score interchangeability. A linkage of this type would not necessarily be generalizable to past or future students writing either assessment. Rather, the linkage would be relevant only to specific administrations of the two assessments, those referred to in the estimation of the linkage.

In this type of linking situation, the reference population used for construction of the regional assessment is expected to represent a small fraction of the reference population for the inter-regional assessment. However, the regional reference population is not necessarily a subset of the inter-regional reference population. In many cases there will only be an intersection between the two groups. For example, many inter-regional assessments are age-based, while regional assessments tend to be grade-based. The sub-population of students who might be expected to take both, the *linking population*, would only be those students of a certain age in a certain grade. Moreover, while the assessments may

be assessing the same broad cognitive domain, how this domain is operationalized in terms of test specifications may differ greatly. For example, a provincial assessment might assess skills defined by the expected learning outcomes, while an inter-regional assessment is more likely to target general skills. The expected learning outcomes are achieved through mastery of the instructed curriculum, while many of the general skills may be learned out of a school context. As noted above, the statistical methods used to produce individual scores based on item responses may also be very different, resulting in somewhat arbitrary score distributions.

Linear Scale Linking Methods

Previous studies linking specific regional and inter-regional assessments, primarily involving NAEP, have used a variety of methods to map the distributional characteristics of X onto Y and vice versa. The simplest method is linear linking, a regression-based method which adjusts the mean and standard deviation of one assessment so that they are equal to the mean and standard deviation of the second assessment. Linear linking of regional and inter-regional assessments was performed by Beaton and Gonzalez (1993) for the 1991 International Assessment of Educational Progress (IAEP) results for the U. S. sample of 13-year-olds and the 1990 NAEP data for public school students in Grade 8. The linking function they estimated was used to compare projected NAEP performance between individual states and 15 countries. In general, performance of the majority of the States fell below the projected performance of most countries in the study, with the notable exception of Jordan, which had a very high proportion of low-achieving students.

Linear linking provides valid results when the score distributions to be linked are expected to have similar shapes. The simplicity of the linear linking functions make them the most invariant to sampling error, because there are only two parameters to be estimated from the data, the mean and standard deviation.

When the distributions are similar in shape, the score distributed will be approximately equal for the target and the transformed scores. On the other hand, if the distributions are unequally skewed, the linear linking function will produce biased results. Score equivalents will be systematically over and under estimated, depending on if they are higher or lower than the sample score mean.

Other linking methods based on linear regression have been used to establish links between NAEP and other assessments. Pashley and Phillips (1993) investigated linking mathematics performance on the 1991 IAEP to performance on the 1992 NAEP. In 1992, they collected sample data from 1,609 U. S. students who were administered both instruments. A simple linear regression model based on this linking sample was used to predict the NAEP performance of students in other countries participating in IAEP, based on projected country performance and individual IEAP performance. McLaughlin (1998) developed more complex regression-based linear models for linking NAEP to statewide assessments, using the 1996 state NAEP Grade 4 and 8 mathematics assessments in four states. McLaughlin's models increased the accuracy of predicting the results of examinees across the assessments through the use of *collateral information* -- other information available describing examinees. The linking sample was defined by students who had written both statewide assessments and NAEP. Using individual student state assessment results, combined with information about their school mean performance and demographic characteristics, the multivariate regression models produced acceptably accurate projections of group-level NAEP scores and percentiles in terms of matching actual NAEP performance results. However, the linking functions estimated in one year did not provide accurate projections for other years.

These methods share a limitation common to all regression analyses. Linear regression predicts an outcome (in this case, the score on an assessment) using other variables which are known to have a relationship with the outcome (in this case, the score on a different assessment as well as personal characteristics of

each examinee). As the number of predictors used in the model increase, the ability to accurately predict the outcome for any individual in the sample also increases. However, as the accuracy of prediction for individuals in the sample increases, the generalizability of the model to individuals outside of the sample decreases. The goal of the linking procedure is to make inferences for individuals outside of the linking sample, which means that the validity of linear regression-based linking methods decreases as the amount of collateral information used increases.

Equipercentile Linking Methods

Without requiring collateral information, the assumptions underlying estimation of a linear linking function can be generalized in order to link scores with different shaped distributions. The primary assumption of linear linking methods is that, across repeated measurements, the expected score and expected deviation of the first test score distribution can estimate the expected score and expected deviation of the second test score distribution. It follows that, across repeated measurements, the expected score of any individual on the first test should estimate his or her expected score on the second test. The percentile rank of any individual should remain constant across assessments, as percentile rank is a function of the distribution of the underlying latent trait in a given population, rather than the scale of a particular assessment. Therefore, the scores corresponding to a particular percentile rank on different assessments can be used to estimate each other. This method is known as *equipercentile linking*. A complete development of the equipercentile linking function is described in Braun and Holland (1982), where it is described as *equipercentile equating*.

With an equipercentile linking design, the transformed score distribution has identical distributional characteristics as the original distribution. This includes the mean and standard deviation as well as higher central moments, such as skewness and kurtosis. When the shapes of the distributions are identical,

equipercentile equating, in the absence of random error, will produce the same results as linear equating. However, the equipercentile procedure is much more sensitive to violations of equity between the two assessments. Ercikan (1997) evaluated the stability of equipercentile linking of test forms published by CTB MacMillan/McGraw-Hill to the 1990 NAEP mathematics scale across different populations. Four states that participated in the 1990 Trial State Assessment of Grade 8 mathematics were included in the study. One of the four states used Form E of the California Achievement Tests (CAT), while the other three used Form E or Form 4 of the Comprehensive Test of Basic skills (CTBS). The CAT and CTBS scores were converted to a common scale, and the equipercentile linking functions were estimated. Six different links were estimated using the available data. Within-state links were estimated for each state, and an overall link was estimated using the combined data from all four states. In addition, a link was estimated using the combined data from the three states using the CTBS. The results of the six linking functions varied considerably more than would be expected due to sampling error.

Linn and Kiplinger (1994) investigated the stability of equipercentile linking of state-level test results to NAEP. Their study used four states that participated in the 1990 and 1992 Grade 8 Trial State Assessments of mathematics. Equipercentile links were established that projected NAEP equivalents for statewide assessment scores using 1990 results, and the linking function estimated was applied to 1992 state assessment results to predict NAEP performance. The predicted results were then compared to the actual 1992 NAEP results for each state. The predictions were reasonably accurate around the median, but less so at the tails of the distribution. Separate equating functions were also developed for male and female students for the two states where gender identification was available from the state test data. Their findings suggest that these methods should be used for approximations only.

The results from the 1996 NAEP and the 1995 Third International Mathematics and Science Survey (TIMSS) assessments at Grade 4 and Grade 8 were linked in a similar manner (Johnson & Owen, 1998). A comparison of linked Grade 8 results with actual Grade 8 results from states that participated in both assessments suggested that the link was working acceptably. However, results for Grade 4 did not produce satisfactory results, as the predicted distribution of test scores did not correspond well to the actual distributions.

The variability of the results of equipercentile methods in these studies underscores the importance of equity to the validity of linking results. The underlying assumption of equipercentile procedures is that the underlying or latent trait responsible for test performance is distributed equally across assessments. Hence, any differences in observed distributions are the result of the arbitrary scale of each test. However, when performances on the different assessments are the result of differently distributed latent traits, the equipercentile linking function is no longer valid. As noted by Ercikan, 1997, this problem may occur when the balance of content within the assessment domain is different between assessments. Differences in the distribution of the latent trait may also occur when a systematic change has occurred over time, such as a non-equivalent grade cohort, or a change in curriculum or instructional policy. Possibly for this reason, many studies have found that equipercentile functions used to estimate a projection linkage did not necessarily generalise across years or groups.

Another reason for the instability of equipercentile methods may be the inherent inaccuracy of discrete percentile coordinates to represent the continuous nature of the latent trait distributions. There are two sources of error in using percentile coordinates: the discrete nature of the observed or assigned scores, and the discrete number of sample points at which percentile coordinates are calculated (Braun & Holland, 1982; Kolen & Brennan, 1995, Chapter 2). As the number of possible scores or the number of sample points decreases, the less accurate will be the equipercentile linking function. Even though the number of

sample points may be increased infinitely, the linking function will still be linear between observed score values, which limits the accuracy of the linking function. For any given sample, the proportion of examinees at a certain score value estimates the population proportion of examinees with that score value. However, particularly with scales derived using IRT, the number of observed scores in a sample is much smaller than the number of possible scores for a given test, and the linking function must be interpolated across scores that were not observed. In order to reduce the errors introduced by interpolation, percentile coordinates sampled from continuous score distributions, which are estimated using all the available data, are used rather than interpolation of the two adjacent observed scores. Braun and Holland (1982) refer to the process of transforming a discrete distribution into a continuous function as *continuisation.*

In the majority of regional/inter-regional linking situations, the sample distribution is not likely to be equivalent to the distribution for the full reference population of either assessment. Regardless of the scale properties (interval or quasi-interval) of the assessment, the scores comprising the linking sample distribution will be missing significant sub-populations. For inter-regional assessments, this deficiency is expected, because there is typically a significant amount of inter-regional variation in educational systems (Bryk and Raudenbush, 1992). The sample is also unlikely to be randomly equivalent to the regional assessment population, because regional assessments are typically grade-based, and national and international survey assessments are typically age-based, which was the case for the Beaton and Gonzalez (1993) study. Even though the reference population is usually scaled to have a normal distribution of scores, removal or selection of specific sub-populations for the linking sample will likely result in a non-normal distribution of scores. Thus, these linking situations require a general method for linking assessments with expected score distributions that are non-normal and of different shape.

However, given the variability of results of equipercentile methods in previous studies, it is clear that non-linear methods need to be used with caution. In particular, attention must be paid to the theoretical equivalence of the underlying trait presumed to be responsible for test performance and the distribution of that trait across assessments. While the limits of interpretability and generalisation are established in most projection or moderation situations, the sensitivity of equipercentile links to sample characteristics suggests that a compromise needs to be found between the sample invariance of linear methods and the sensitivity of discrete equipercentile methods.

PURPOSE OF THE STUDY

The purpose of the current study is to examine methods of estimating functions that will be able to link assessment scales in situations where the observed score distributions of the linking sample are expected to be non-normal and dissimilar in shape. The methods will be evaluated as general solutions to the problem of linking information from regional and inter-regional assessments under the following conditions:

1. the assessments are intended to measure the same general cognitive domain;
2. there will be a linking sample(s), such that the sample taking Assessment X is assumed to be randomly equivalent to the sample taking Assessment Y. This may occur where the same sample writes both assessments or where assessments are administered to different random samples drawn from the linking population;
3. the linking sample(s) may not be representative of the populations used to calibrate/norm the original tests;
4. the methods used to calibrate, norm, or standardise the two tests may be dissimilar, and
5. transformed scores will be reported only at aggregate levels.

METHOD

DATA

Instruments

The data considered in the present study were provided by the Foundation Skills Assessment (FSA) and the Programme for International Student Assessment (PISA). The Foundation Skills Assessment (FSA) component is a provincial examination administered to students in British Columbia studying in school at the Grade 10 level, excluding special education students (British Columbia Ministry of Education, 2000). In 2000, the FSA was administered to approximately 48,000 Grade 10 students. A common test form with three components was used, and the students' responses were used to produce scores separately in reading, mathematics, and science. The relationship between expected performance on an item and examinee proficiency was described by a generalized partial credit model (Muraki, 1990), and Bayesian estimates of examinee proficiency were produced using item responses and a single prior describing the empirical population density. These methods were implemented with the software PARSCALE (Muraki & Bock, 1997). The final scores were reported at an aggregate level, with a provincial mean of 500 and standard deviation of 100, three levels of performance: Not Meeting Expectations, Meeting Expectations and Exceeding Expectations. The respective score cut-points for assigning students to these performance levels were 432.91 and 636.88.

The Programme for International Student Assessment (PISA) is an ongoing international survey administered to nationally samples of students representing 15-year-olds attending school in 32 member countries of the Organisation for Economic Co-operation and Development (OECD). In PISA 2000, the assessment was a balanced incomplete block design, with 10 separate booklets. The items were calibrated using the mixed coefficients multinomial logit model (Adams, Wilson, & Wang, 1997). The calibration and equating of

items and booklets was performed using a synthetic international population of 13,500 15-year-olds. This sample was composed of 500 examinees from each of 27 participating OECD countries that met the PISA 2000 response rate standards, stratified on linguistic community within each country and excluding special education students.

The student responses were scaled to simultaneously produce scores in reading, mathematics, and science with an international mean of 500 and standard deviation of 100. Individual estimates of proficiency were produced using a hierarchical Bayesian technique, where the location of the conditioning (*prior*) distribution was determined by multivariate regression using information from approximately 280 variables available from the accompanying PISA student questionnaire (Adams, Wilson, & Wu, 1997; *PISA 2000 Technical Report,* 2002). Essentially, the multinomial logit model increases the precision of proficiency estimation by considering all other information available on each student, in addition to the information available from the test items. The resulting posterior distribution for each examinee was represented by five random draws, or *plausible values,* selected according to the probability defined by his or her individual posterior score density function. Thus, instead of using a single point estimate to represent performance in secondary analyses, the five plausible values are used simultaneously. The derivation and use of plausible values in the context of a large-scale complex survey assessment are presented in Mislevy (1991). The error of measurement for each individual is proportional to the variance between the five plausible values. Each plausible value, while inaccurate at the individual level, accurately reproduces distributional parameters of groups. Similar to the FSA, results of PISA were reported at an aggregate level as proportions of students with scores corresponding to specific levels of performance. PISA used five performance levels, with the following cut-points: 335, 308, 481, 553, and 625. Students with scores below 335 (below Level 1) were not assigned a specific performance level, because performance was defined by positive descriptors.

Content Equity

The FSA does not assess curricular achievement with respect to any single course or grade (British Columbia Ministry of Education, 2000). However, the skills assessed are defined by the general learning outcomes overarching the reading curriculum in British Columbia. The focus of PISA is similar – it is designed to assess the outcomes of educational systems, rather than mastery of specific curricular objectives (OECD, 2000). The PISA reading literacy domain was defined by three types of tasks: *retrieving information*, *interpreting*, and *reflecting/evaluating*. The FSA reading comprehension domain was also defined by three types of tasks: *identifying and interpreting key concepts and main ideas*; *locating, interpreting and organizing details*; and *critical analysis*. Both assessments required examinees to read a variety of textual stimuli in different forms and for different purposes.

A qualitative evaluation of content equity between the FSA and PISA reading components was performed by three content experts provided by the Student Assessment Branch of the British Columbia Ministry of Education. The Student Assessment Branch was responsible for the development, standard setting, and scoring of the FSA. In addition, the Student Assessment Branch participated in PISA item selection and was responsible for overseeing the administration of PISA in British Columbia. Each of the reviewers had previous experience in some or all of these aspects of both assessments. The results of this evaluation are summarized in Cartwright, Lalancette, Mussio, & Xing (in press).

Overall, the two assessments were found to be assessing the same general domain of reading proficiency, with approximately the same balance of conceptual difficulty. However, PISA used stimuli were more likely to be found in non-scholastic settings, while FSA included more works of literature, such as narratives and poetry. In the absence of a common framework, items from each test were matched to the content framework for the alternate assessment and

compared. Table 1 summarizes this comparison. The balance of tasks using the FSA framework was very similar, with the majority of items requiring *locating, interpreting and organizing details.* Using the PISA framework, FSA had approximately the same proportion of *interpreting* items, but the reviewers could not identify any *reflecting/evaluating* items. As a result, the proportion of *retrieving information* items was much higher in FSA than PISA.

Table 1

Comparison of Item Content Representation in FSA and PISA as a Percentage of Total Items.

	PISA (%)	FSA (%)
FSA Category		
Identify/Interpret Key Concept & Main Ideas	12.2	9.3
Locate, Interpret and Organize Details	51.7	55.8
Critical Analysis	36.1	34.9
PISA Aspects of Reading		
Retrieve Information	29.4	51.2
Interpret	50.3	48.8
Reflect and Evaluate	20.3	0.0

Source: Cartwright, Lalancette, Mussio, & Xing (in press)

The similarity of these two assessments in terms of content and purpose was quite close, given that they were developed independently. In addition, they are very similar in terms of measurement precision (see Properties of Scores Distributions and Precision, below). This similarity suggests that the FSA-PISA linkage is simply a calibration of the two scales, with the correspondingly high validity of inference. However, the scales used to interpret performance on the assessments are subject to many factors that change significantly over time. For

example, changes in curriculum or school policy in British Columbia or any of the countries participating in PISA will systematically change relationship between the latent distributions of proficiency in the two calibration populations. Therefore, the link that is established should be interpreted as between projection and calibration, with a fixed reference to the 2000 administrations of FSA and PISA. The limitations of a projection linkage are within the parameters for regional and inter-regional linking described in the purpose of the current study.

Sample

In 2000, the PISA assessment was administered to 3037 15-year-old students in British Columbia. The majority of these 15-year-olds were in Grade 10. Most of these 15-year-olds in Grade 10 were also administered the FSA in 2000. The number of students who wrote both the FSA and PISA in 2000 was 2801. Administration of the FSA was scheduled to be between May 1 and May 5, 2000, and the administration of PISA was scheduled between May 15 and June 2, 2000. Both FSA and PISA assessed student performance in multiple domains. However, mathematics and science were minor components of PISA 2000, and only five-ninths of the PISA sample was assessed in these subjects. Furthermore, the number of items used in PISA science and mathematics was less than a quarter of those for reading, resulting greater measurement error. Therefore, due to sample limitations of both examinees and test items, only the reading components were considered in the present study.

Properties of Score Distributions

As noted above, the observed score distributions are not equivalent, because the populations on which the scores were scaled are different. The PISA assessment is scaled across a sample from a synthetic international population of 32 equally represented countries, whereas the FSA assessment is scaled across a near-census of the actual population of British Columbia Grade 10 students.

Canada (and particularly Grade 10 students in British Columbia) performed well compared to other countries on PISA, and is close to the upper bound of the PISA assessment. As a consequence, the British Columbia PISA reading score distribution is negatively skewed. Although not to the same extent, there is a similar issue for the FSA, because the sample of 15-year-olds writing PISA who also wrote FSA is not a fully representative sample of the population of Grade 10 students about whom FSA scores generalise. Although both tests were scaled to have a normally distributed population distribution of scores with mean and standard deviation of 500 and 100, respectively, the systematic differences between the two reference populations results in different score distributions.

The four empirical shape parameters of the linking sample for both the FSA and PISA score distributions are shown in the first and seventh rows of Table 1. Although the difference in means does not imply any difference in the shape of the score distributions, the smaller standard deviation of the PISA scores indicates that, overall, examinees appear more different from one another on the FSA scale than on the PISA scale. The PISA scale also has a greater negative skewness, which suggests that examinees with high scores tend to appear more similar on the PISA scale than they do on the FSA scale, even after accounting for the difference in standard deviations. Similarly, the smaller kurtosis of the FSA score distribution suggests that examinees with scores around the average appear to be more similar on the FSA scale than on the PISA scale, even after accounting for the previously described differences in scale properties.

Equity in Measurement Precision

Although projection or moderation linkages, previously described, do not require equity in measurement precision, the validity of the linkage increases if the two assessments have similar precision. Measurement equity was examined by comparing the error of measurement of the two tests to their respective standard deviations, using the final linking sample (see below). Using Equation 1,

taken from Crocker and Algina, (1986, p. 352), it was possible to estimate the classical index of reliability, η, for each test.

$$\eta_{J\theta} = \sqrt{1 - \frac{\sigma^2{}_E}{\sigma^2{}_J}} \qquad (1)$$

where σ^2_E is the average measurement variance across the sample, and σ^2_J is the variance of all scores in test J. The average variance error of individual scores in the linking sample for the PISA test is 958.19, and for FSA, 1003.36. The variance error of measurement of individual cases was calculated for the PISA plausible values as:

$$\text{var}_{measurement}(pv_1, pv_2, \ldots, pv_M) \approx \left(1 + \frac{1}{M}\right)\frac{1}{M-1}\sum_{i=1}^{M}\left(pv_i - \overline{pv}\right)^2, \qquad (2)$$

where pv_i represents the ith of M plausible values, and $\overline{pv}$ represents the average of all M plausible values (adapted from Mislevy, 1991). Given the sample standard deviations of 91.1 and 92.6, both instruments had classical reliability indices of 0.94. These results suggest that, for the purposes of this project, the assessments possess equity in measurement precision.

PROCEDURE

Necessary Assumptions for Estimating the Linking Function

As described earlier, the basic assumption of equipercentile linking is that the expected score of an individual, i, on one assessment, $E(x_i)$, estimates his or her expected score on another assessment, $E(y_i)$. The equivalence between the scores exists because the expected scores are functions of the scales of the assessments, X and Y, which are functions of a common latent trait governing test performance, θ_i. Therefore, any distribution of observed scores can be expressed in this manner, as $P(X(\theta))$. Furthermore, the distribution of scores is conditional on the distribution of θ, rather than any specific θ_i. Consequently, in estimating an equipercentile linking function, it is sufficient to assume that the distribution of

the latent trait, P(θ), is equivalent across the observed score distributions. In any pair of randomly equivalent samples (i.e., randomly drawn from the same population), the distribution of the latent trait will be equivalent, and the expected score for an individual at a certain percentile rank in one sample estimates the expected score for an individual at the same percentile rank in the other sample.

In order to estimate a linking function between regional and inter-regional large-scale assessments, an additional assumption is required in order to deal with the gap in time between administrations of the two instruments. Large-scale assessments, in order to provide accurate information about a population, are subject to specific time constraints determining when they can be administered. Typically, provincial assessments occur at the end of a school year or semester, after a period of formal instruction. In Canada, this is usually during the months January and June. For school-administered inter-regional assessments, administration windows are chosen to minimize the disturbance to classrooms. In Canada, the administration of inter-regional assessments, such as SAIP and PISA, takes place between the months of February and June. Given that these instruments assess traits that are presumed to develop with time, the time gap between administrations may introduce a systematic difference in the distribution of θ. Previous longitudinal research into the development of cognitive skills suggests that, as students age, the distribution of a cognitive skill remains normally distributed, but with an increasing mean and standard deviation (e.g. Muller, Stage, & Kinzie, 2001). Thus, it is reasonable to assume that the latent trait of any examinee at a second testing period, θ_2, is equal to his or her latent trait at the first testing period, θ_1, plus the average change in the latent trait across the population, c, and a normally distributed residual, r, with a mean of zero and a variance of σ_i^2. This relationship can be expressed as:

$$\theta_2 = \theta_1 + C + R, \qquad (3)$$

where $R \sim NID(0, \sigma_i)$,

with C and σ_t increasing as the length of time between examinations increases (the same equation holds for random samples from the same population over time). A significant feature of this equation is that the transformation is linear, which means that the basic shape of the latent distribution remains the same. As a consequence, if the exact same instrument were administered twice, the observed score distribution of the second administration will be a function of a linearly transformed latent trait distribution:

$$P(Y(\theta_2)) = P(Y(\theta_1 + C + R)), \quad (4)$$

indicating that the score distribution at the second administration will be wider and positively translated.

In the current situation, where the goal is to estimate a linking function, G, such that:

$$P(Y(\theta_2)) = P(G(X(\theta_1))), \quad (5)$$

substituting for the relationship described in Equation 4 produces a complete description between the score distributions of X and Y, with respect to θ:

$$P(Y(\theta_2) = P(G(X(\theta_2 - C - R))). \quad (6)$$

Given that administration times are fixed for the types of large-scale assessments examined in the current study, the systematic change in the distribution of θ can be considered an essential property of the scales themselves, because the scores have a specific temporal reference. Thus, the estimation of linking function, G, assumes that the systematic differences introduced by the time between administrations are properties of the scales and will be accounted for in converting scores from scale X to scale Y and vice versa.

Estimation of the Equipercentile Linking Function

The linking problem in this study is equivalent to equipercentile observed-score equating based on estimated true score distributions (see Kolen and Brennan, 1995, pp. 67-83). Subsequent equations in this paper represent the

scaled value of θ_x as x, and the scaled value of θ_y as y. The steps involved in equipercentile linking are:

1) estimate cumulative density functions (CDFs) for each test score distribution

$$F(x) = Pr(X>x), \quad (7)$$

where x is a possible test score from the distribution of test scores, X.

2) find inverse CDFs, $F^{-1}(x)$, for each test to predict the observed score given the cumulative distribution (i.e., percentile rank), such that

$$F^{-1}(F^{-1}(x)) = F(x); \quad (8)$$

3) use the CDFs and inverse CDFs to find the equipercentile linking functions

$$e_y x = G^{-1}(F(x)), \quad (9)$$

where $e_y x$ is the equipercentile equivalent of a score on scale Y, given a score x and G is the linking function. Equations 7, 8, and 9, applying to X and x, also apply to Y and y.

The estimation of Equation 7 can be carried out explicitly, but this will produce a "jagged" CDF that is composed of linear segments joining CDF values for each observed score. This empirical CDF, although perfectly representative of the given data, is likely to introduce a substantial amount of sampling error. It is believed that the underlying continuum of proficiency represented by the test scores is continuous; therefore, any function describing the distribution of scores is expected to also be continuous. Any departures from continuity are assumed to be a result of our inability to sample individuals or measure scores in the missing intervals. However, an examination of the derivative of the empirical CDF will reveal one greater discontinuities than there are observed score values. If a better sample had been taken or finer measurement precision were available, presumably the missing intervals would be "filled in" to produce a non-jagged function. Typically, the sample is limited and the instruments are fixed, so the observed CDF must be "smoothed" to produce a continuous CDF. Unfortunately, the steep slope of CDFs around the point of inflection makes the linking function extremely sensitive to arbitrary choices in smoothing. However, since

$$\Pr(X \geq x) = \int P(x)\partial x , \qquad (10)$$

Step 1 becomes a problem of estimating of continuous cumulative density functions, for which there are many well-known methods, typically based on normality or uni-modality assumptions. Much work has been conducted, particularly at the American College Testing Program (ACT), in developing methods for estimating continuous score distributions (e.g., Cope & Kolen, 1990; Hanson, 1990). In general, the findings suggest that the most accurate method involves the use of the family of 4-parameter Beta distributions (described below). However, the use of the family of 4-parameter Beta distributions is restricted to situations where the only deviations from normality in the theoretical score distribution are skewness and kurtosis. Unfortunately, given the non-representative nature of the linking sample(s) noted above and the possibility of non-interval score scales, the distribution of the scores is expected to be non-normal, and possibly multi-modal, for one or both of the assessments. In view of the fact that the true score distribution is unknown, there is no way to know, in estimating the continuous distributions, which fluctuations are real and which are random. Minimising the influence of the sample fluctuations minimises variance error, but also increases bias (mean error), resulting in a loss of distributional information (see Hastie, Tibshirani, & Friedman, 2001, for a comprehensive explanation of the bias-variance trade-off in a variety of statistical estimation procedures). Ultimately, minimising the effects of random error leads to simple linear linking procedures, which ignore score distribution moments higher than the second order. Alternately, minimising bias by making the procedure very sensitive to data fluctuations may increase variance error, because most of the data fluctuations are likely to be random. The goal of the estimation procedure is to minimise the combined error from both model bias and random sources by increasing sensitivity to real sample fluctuations while decreasing sensitivity to random sample fluctuations.

Consequently, two alternative strategies were developed for this project. Together with the 4-parameter Beta family, they represent a spectrum of reasonable MSE-reducing approaches: Gaussian kernel smoothing, which is the most data-sensitive; finite Gaussian mixture models, which represent a mix of data sensitivity and model-dependence; and four-parameter beta distributions, which are the most model-dependent. The trade-off between flexibility to describe the data accurately and insensitivity to random error is reflected in the degrees of freedom of each model to fit the data. Although it is essentially non-parametric, the number of "parameters" (i.e., kernel weights) estimated with kernel smoothing is data dependent and thus has the potential to be as high as the number of observations. The number of parameters for the mixture models will be a factor of the number of mixture components. The beta model will have only four parameters, giving it the most degrees of freedom and thus the least sampling variability. Each of these methods is described in further detail below.

The integrals of these continuous functions are used, by definition, as the cumulative density functions, which are matched using equipercentile methods. The complexity of these functions makes it unlikely that a real algebraic inverse function exists, so quadrature methods are used to approximate the inverses by sampling the functions at several thousand locations. Linear-interpolation of these sample points can then be used to find the final linking function shown in Equation 9.

Influence of Measurement Error

Measurement error and its treatment during score estimation influence the estimation procedure described above by altering the properties of the observed score distributions. Measurement error in scale scores typically increases dramatically at the extremes for tests of finite length (Lord, 1980), reaching a maximum for examinees with uniformly correct or incorrect responses. The most likely proficiencies to produce perfectly correct or incorrect (uniform) response

vectors are infinitely high or low, respectively, yet it is extremely unlikely that any examinees truly have infinitely high or low proficiencies. Therefore, IRT estimation of proficiency requires the specification of arbitrary maxima and minima that usually correspond to two or three standard deviations from the mean. All examinees with extreme responses receive the same arbitrary high or low scale score.

This method, although sufficient for providing individual estimates, introduces bias at the distribution level. As a consequence of test construction; there are typically a greater proportion of examinees with uniform response vectors than would be predicted by the area above or below two or three standard deviations on the normal curve. For a fixed length assessment designed to provide the maximum information for the most examinees, there is little benefit in including items that provide a lot of information at the extremes of proficiency. Thus, beyond certain reasonable limits of extreme proficiency, all extreme cases have a relatively equal and high probability of achieving a uniform response vector. The clustering of all of extreme examinees at two score values produces artificial modes at the extremes, although it is expected that, had the test more information at extreme ranges of proficiency, these extreme modes would not exist.

Many strategies exist to increase the accuracy of individual score estimates, and thus reduce the errors introduced at the distribution level. As previously described, the scaling procedure used for PISA involves a method of reducing measurement error through the use of collateral information provided by examinees and proficiency estimates on correlated skill domains. As a result, PISA respondents with perfectly correct or incorrect scores may still have a large number of possible scores, depending on their other individual characteristics. Combined with the random imputation method described in Mislevy (1991), this method produces five plausible values – individually inaccurate point estimates of proficiency that produce unbiased and efficient estimates of the population

distribution of proficiency. Because the five plausible values represent the posterior density of each individual, the use of the full set data points, equal to five times the number of cases in the sample, will allow more accurate estimation of the continuous population score distribution than using each set individually.

For the FSA, although population characteristics are used to decrease measurement error, only a single prior distribution is applied to all scores, and the expected a posterior estimate produces a single score for each examinee. Thus, there are still artificial modes at the extremes of the FSA score distribution. These extreme scores on FSA do not share the same distributional properties as the rest of the FSA scores, and their frequency is an artifact of an arbitrary treatment of measurement error. The linking methods used examine population distributional properties; therefore these scores must be removed prior to applying a linking procedure. In order to prevent undue influence of extreme FSA scores on the linking procedure, cases with scores of 0 and 791. 07 on the FSA (the minimum and maximum observed scores, respectively) were removed from the sample used for this project, resulting in a common sample of n = 2659 students.

Step 1: Estimating Cumulative Density Functions

As noted above, the first step simplifies into the more general problem of estimating a probability density function for the random variables X and Y, representing continuous score distributions on the two tests. Four different methods of estimating continuous distributions were compared: (1) Gaussian kernel estimation; (2a) finite mixture models with variable number of components; (2b) finite mixture models with fixed number of components; and (3) 4-parameter beta distributions. The programs used for estimation of each model are presented in the Appendix with additional details on each specific algorithm.

Gaussian Kernel Density Estimation

The Gaussian kernel density estimation method involved in the present study used estimates of the population distributions of scores as the mean of J kernels centred on each of the j observed scores in the sample (Härdle, 1990; Silverman, 1986). Thus, the population density function, $P(x_i)$ is defined by:

$$P(x_i) = \sum_j K(x_i | x_j) p(x_j), \quad (11)$$

$$K(x_i | x_j) = e^{-2^{-1}(x_i - x_j)^2 (h^* x_j)^{-2}}, \quad (12)$$

$$p(x_j) = \frac{\sum_{i=1}^{n} r(x_i | x_j)}{n}, \quad (13)$$

$$r(x_i | x_j) = \begin{cases} 0 & if\ x_i \neq x_j \\ 1 & if\ x_i = x_j \end{cases}, \text{ and} \quad (14)$$

i=1,2,…, i, j, …n,

where the kernel function, $K(x_i | x_j)$ is summed across all j unique observed scores weighted by their probability of occurrence, $p(x_j)$ and n is the sample size. The kernel function (Equation 9) used is the Gaussian normal density function, with a parameter, *h*, defining the "spread" of the kernel. Larger values of *h* result in a wider kernel, and thus produce a greater amount of smoothing. As *h* increases, the sampling error decreases, but the estimated distribution becomes more uniformly flat, resulting in increased bias.

Given the nature of the score distribution, it is likely that the distance between adjacent observed scores may differ significantly, depending on where the adjacent scores in the population density. For example, around the average, adjacent scores may be quite close, because the probability of a score being in that range is very high. However, extreme scores are much rarer, and the distance between adjacent extreme scores may be a tenth of a standard deviation or greater. For this reason, it is appropriate to vary the bandwidth parameter, *h*, in the above kernel function to reduce "bumpiness" in the estimated density function. The

optimal kernel used for this analysis is the "Better Rule of Thumb" bandwidth defined by Härdle (1990, p. 91) as

$$h = \frac{4}{3} Min\left(\sigma, \frac{R}{1.34}\right) n^{-\frac{1}{5}}, \qquad (15)$$

where σ and R are the standard deviation and interquartile range of the observed scores, respectively. Cope and Kolen (1990) suggested that varying the kernel function for each observed data point could increase the accuracy of a kernel estimator. This adjustment allows the kernel to provide more smoothing where scores are sparse, while avoiding over-smoothing where the score distribution is denser. Variable kernels are described in Silverman (1986). The optimal kernel described above was varied according to the k^{th} root of the mean distance to the nearest neighbours for each observed score. The value of k was determined based on comparison of several values -- in general, higher roots will result in a more uniform kernel, and lower roots will result in a more varied kernel. This adjustment produced the following function for h^*_{xj} for score x_j:

$$h^*_{xj} = h\left(\frac{x_{j+1} - x_{j-1}}{2}\right)^{\frac{1}{k}} \qquad (16)$$

In order to speed the algorithm, the density was defined discretely at 2000 sample points evenly distributed between arbitrary endpoints. Silverman (1986) recommended endpoints equal to the maximum and minimum observed scores, plus and minus $3h$. These values were used in the present study. Finally, a piecewise linear function was interpolated between the sample points and divided by its definite integral, limited by the endpoints defined above, to produce the final density estimate, $P(x_i)$.

Finite mixture models

The essential premise of a Gaussian mixture model is that any observed non-normal distribution can be described by a finite number of normal distributions. In many educational situations, this type of clustering is expected,

given the influence of factors like institutional effectiveness and subpopulation membership on performance (Bryk & Raudenbush, 1992). In a most basic example, a distribution of test scores may have two distinct modes, corresponding to two systematically different subpopulations. If it were possible to determine which individuals belonged to which subpopulation, then it would be possible to estimate the parameters describing the normal distribution of each subpopulation separately. Adding the two components, weighted by their relative size develops a parametric model of the overall population distribution. In more complex cases, where there are many distinct subpopulations, each normally distributed, it would be necessary to estimate the distribution for each subpopulation separately. Two types of mixture models were investigated in the present study: variable component models and fixed component models.

Variable component mixture models

The full variable component model takes the form

$$P(x|\, W,M,S)_j = \sum_{i=1}^{k} w_i \left[\frac{1}{s_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(s_i^{-1}(x-m_i)\right)^2} \right], \qquad (17)$$

where W is the vector of weights for the components, M is the vector of locations for the component, and S is the vector of standard deviations for each normally distributed component. The W, M and S vectors are all of length k, the number of components used in the mixture.

In practice, it is rare that the subpopulation membership is known for each individual. Furthermore, the number of subpopulations is also unknown. Thus, the estimation of a finite mixture model uses a maximum likelihood approach to estimate the parameters for each model with a specified number of components, as well as to evaluate estimated models with different numbers of components. The estimation of model parameter vectors (W,M,S) from a given number of components follows the Expectation-Maximization (EM) algorithm for finite mixtures (see, e.g., Dempster, Laird, and Rubin, 1977; Everitt and Hand, 1981).

Although each EM cycle always converges on the maximum likelihood estimates, the convergence is extremely slow. The Newton-Raphson procedure for finding maxima and minima (see, e.g., Everitt & Hand, 1981), while much faster at converging, will often diverge if given poor initial estimates. The feasibility of combining the two methods was investigated, by using the EM algorithm to estimate starting values for the Newton-Raphson procedure. The computational algorithms developed for this combined procedure are presented in the Appendix.

In order to determine the optimal number of components, several models, with number of components increasing from 2 to 5, were estimated separately. The final model chosen was the model with the greatest likelihood of producing the data, regardless of the number of parameters estimated. This approach was adopted over the likelihood ratio because, given the sample size, the degrees of freedom would be effectively constant with each additional three parameters.

Fixed Mixture Models

The family of fixed mixture models is identical to the variable-component method of Equation 17, with the exception that the number and location of the distributions is fixed to an arbitrary maximum. That is, the elements, m_i, of M for the fixed mixture models were defined by

$$m_i = (i+1)\frac{\text{Range}(X)}{k+1} \qquad (18)$$

The remaining parameters (component variances and weights) are estimated using the combined EM algorithm and Newton-Raphson procedure used for the variable component models. This approach represents a compromise between the variable mixture model and the kernel density estimator. Although the parameter estimation process is mathematically identical, the conceptual focus shifts from estimating the parameters of each individual component to producing a parametric maximum likelihood model of the full sample distribution. Thus, even though the locations of the components are fixed, any observed modality in

the data can be modelled by changing the relative weightings and variances of adjacent components. A noted problem with estimating mixture models using iterative techniques is that there is always the possibility of components becoming singularities (Everitt and Hand, 1981). That is, as the location of a component converges onto the location of a single data point, its variance will go towards zero, because, theoretically, the most likely model to produce the data is a singular component for each observation. By fixing the locations vector, it may be possible to produce a parametric function that accurately describes the data density without concern for convergence on singularities.

4-Parameter Beta Distributions

The 4-parameter beta model can be viewed as a general form of the normal density function (Vijverber, 2000). It is based on the 2-parameter Beta distribution, which has a lower bound of zero and an upper bound of one. The two parameters define the location, skewness and kurtosis, within the (0,1) domain. The 4-parameter Beta distribution adds an additional two parameters that are used to redefine the minimum and maximum limits of the score domain. The 4-parameter model is defined by

$$g(x|\,\alpha,\beta,l,u)=\frac{(-l+x)^{\alpha-1}(u+x)^{\beta-1}}{(x-l)^{\alpha+\beta-1}B(\alpha,\beta)}, \qquad (19)$$

where $B(\alpha,\beta)$, the beta function, is related to the gamma function, $\Gamma(x) = (x-1)!$ (for $x>0$), by

$$B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}. \qquad (20)$$

The α and β parameters describe the location, skewness, and kurtosis of the function, given the domain. The other two parameters, l and u, describe the lower and upper limits of the domain, respectively. Estimation of the four parameters follows the method of moments algorithm for estimating three parameters of the 4-parameter Beta distribution, when the fourth parameter is specified (described

in Johnson & Kotz, 1970, and amended by Hanson, 1991). The algorithm used in the current study requires specification of the lower limit parameter and estimates the remaining parameters accordingly, using the non-central moments of the sample data. The specified value of the lower limit was chosen to minimise the squared deviation of the higher moments of the fitted score distribution from the higher moments of the sample. During evaluation of the computational algorithm, it was observed that the squared deviation of fitted-to-observed moments approaches zero as the lower limit approaches negative infinity, so the lower limit is chosen as the minimum value that produced real values for the alpha and beta parameters of the distribution (given machine precision). Because the method of moments involves exponents, the value of the lower limit is affected by the scale of the variables – scores with a large number of reported digits and where the majority of scores are far from zero require that a greater number of digits and decimals places be tracked in each computational step. Thus, the specified lower limit approximated negative infinity. In order to determine this value, the algorithm looked for the lowest value to produce real-valued results on an inverse quadratic path, so that, as the lower limit approached the mean of the sample, the incremental changes in the test-values became smaller and smaller.

Step 2: Estimation of Inverse Cumulative Density Functions

The complex form of the equations for the CDFs makes an algebraic solution to the inverse functions computationally unreasonable, even if algebraic inverses were to exist. Therefore, the inverse functions were represented by quadrature methods. Over-sampling at extreme score values was done by selecting sample points according to a non-linear function across a vector, T1, of 4,001 values evenly distributed between -2 and 2 at intervals of 0.001, to produce a vector of score values, vector S1:

$$S1 = \frac{2}{\sqrt{\pi}} \int e^{-(0.75(T1))^2} 5(sd) + \bar{x}\,, \qquad (21)$$

where *sd* and $\bar{x}$ are the estimated sample standard deviation and mean, respectively, of the observed scores. The resulting sample points in S1 had a density roughly inverse to the density of the observed scores. This provided a sufficient number of data points to increase precision for extreme scores. The S1 vector was then used to produce the 4,001x2 matrices C_x and C_y, defined for scale *X* (and equivalently for *Y*) as follows:

$$C_X = [CDF_X(S1) \quad S1]^T, \tag{22}$$

where $CDF_x(S1)$ was a vector of CDF values for *X*, evaluated at each sample point in S1. The process resulted in two inverse CDF matrices for each test, describing coordinates of the inverse function. A piecewise linear function was interpolated from this matrix to approximate the inverse CDF for each scale, CDF^{-1}.

Step 3: Estimation of Score Equivalents

In order to accommodate the piecewise functions of the inverse CDFs determined in Step 2, the score equivalents were also estimated algorithmically. Essentially, the linking function was defined as a matrix of equipercentile equivalents, created by evaluating and matching each inverse CDF at a set of sample points between 0 and 1. The numbers were unevenly distributed, again with the intent to over sample extreme regions. The E matrix, a 4,001x2 matrix which represented score equivalents sampled from the two scales, was constructed with the following formula,

$$E = [CDF_x^{-1}(S2) \qquad CDF_y^{-1}(S2)]^T, \tag{23}$$

where

$$S2 = \frac{1}{\sqrt{\pi}} \int e^{-T1^2} + \frac{1}{2}, \tag{24}$$

and T1 was defined as above. The row elements of E provided 10,001 coordinates used for interpolating a piecewise linear function describing the equipercentile

equivalence of X and Y. This approximated the continuous function to be used for finding score equivalents of X on Y, e_xy.

Computational Considerations

The kernel method and the mixture models required several specifications to be made in order to begin computation. For the kernel method, these were the number of sample points that were used to discretize the score continuum and the value of k, which was used to vary the kernel function. One-thousand sample points were used for this study. Values of k between 2 and 100 were tested, with small differences in results. Values under 10 produced less accurate density estimators, although there was no appreciable improvement as k increased above 20. For the final results of this study, k was set to 20.

For the variable mixture model, it was necessary to specify the maximum number of components, EM cycles, and Newton-Raphson iterations. Several combinations of the two were evaluated. In most cases, the Newton-Raphson iterations were either unnecessary, because the parameters were very near their maximum likelihood values after the EM cycles, or did not converge. For the final estimation, the maximum number of components was set to 5, EM cycles to 1000, and Newton-Raphson iterations to 20 (Newton-Raphson iterations were terminated earlier if parameters did not converge).

For the fixed mixture models, the goal was not to find maximum likelihood component parameters, but simply to find a good fitting function. Thus, by increasing the number of components, it was possible to accurately model the data with a significantly smaller number of machine learning steps. Another reason for reducing the number of parameters was to avoid the singularity convergence described by Everitt and Hand, 1981. The number of components for this study was set at 19, with 50 EM cycles and no Newton-Raphson iterations.

Each of the estimation procedures required one or more passes over the set of observations. For the kernel and mixture model methods, each pass required

large number of computing operations. The time required for each data pass is reduced as the number of unique observations decreases. In order to reduce the time for each data pass, the data were pre-binned to the closest tenth, using conventional rounding rules. In effect, this meant that each score was rounded to approximately the nearest thousandth of a standard deviation. Although this rounding process did not affect the empirical distributional properties of the two sets of scores, it decreased computational time by around one-quarter.

EVALUATION OF DENSITY ESTIMATORS

It is unlikely that a single method of estimation will provide the best means of estimating the population distribution for all scenarios. Kolen and Brennan (1995) recommended that any density estimation procedure should have a statistical framework for evaluating the appropriateness of the model. The methods used in the present study were evaluated primarily on moment preservation and likelihood. Moment preservation examines the agreement between the first four empirical central moments:

$$mean_{emp} = \frac{1}{n}\sum_{i}^{n} x_i \quad , \tag{25}$$

$$variance_{emp}(x) = \frac{1}{(n-1)}\sum_{i}^{n}\left(x_i - \bar{x}\right)^2 \quad , \tag{26}$$

$$skewness_{emp}(x) = \frac{1}{n}\sum_{i}^{n} x\left(\frac{x_i - \bar{x}}{\sqrt{variance_{emp}(x)}}\right)^3 \quad , \tag{27}$$

$$kurtosis_{emp}(x) = \frac{1}{n}\sum_{i}^{n}\left(\frac{x_i - \bar{x}}{\sqrt{variance_{emp}(x)}}\right)^4 \quad , \tag{28}$$

and the fitted central moments:

$$mean_{fit} = \int_{-\infty}^{\infty} P(x)\, x\, \partial x \, , \tag{29}$$

$$variance_{fit}(x) = \int_{-\infty}^{\infty} P(x)\left(x - \bar{x}\right)^2 \partial x, \tag{30}$$

$$skewness_{fit}(x) = \int_{-\infty}^{\infty} P(x)\left(\frac{x - \bar{x}}{\sqrt{variance_{fit}(x)}}\right)^3 \partial x, \tag{31}$$

$$kurtosis_{fit}(x) = \int_{-\infty}^{\infty} P(x)\left(\frac{x - \bar{x}}{\sqrt{variance_{fit}(x)}}\right)^4 \partial x. \tag{32}$$

Closer agreement between the empirical and fitted values indicated that the continuous density estimator captured the distributional properties of the original sample. Lower moment preservation indicated a greater amount of bias in the estimation process. In order to determine the most effective method, the methods were ranked according to their absolute deviation from each of the empirical targets. If a method consistently ranked higher than the other methods across most or all moments, it was considered more effective (likewise, if a method was consistently ranked lower than others, it was considered least effective). On the other hand, if there was no clear order in ranking across all four central moments, no decision could be made.

Each model was also evaluated based on the likelihood that it produced the sample data. Likelihood is estimated based on the definition of each fitted density estimator as the probability function. Because each function, P(x), describes the probability of observing a specific score x_i, this value of P(x_i) is equal to the likelihood that the given datum, x_i, was produced by the function P(x). Furthermore, given a set of data, $x_1, x_2, \ldots x_n$, it is possible to estimate the joint likelihood that the full data were sampled from the specified density, P(x):

$$Likelihood = \prod_{i}^{n} P(x_i|\Sigma), \tag{33}$$

where Σ represents the parameters and functional form of the specific density function being evaluated. For computational simplicity, the natural logarithm of this function was taken,

$$Log - Likelihood = \sum_{i}^{n} Ln\left[P\left(x_i|\Sigma\right)\right]. \quad (34)$$

Higher values of log-likelihood indicated that the given sample was more likely to have been drawn from the given continuous distribution.

A visual comparison of the smoothness of each fitted model provided an overall summary of each method's ability of each procedure to correct for sampling errors while describing the data. As described above, the underlying assumption of smoothness in the population distribution of the latent trait suggests that greater smoothness indicates a more robust density estimate. Variations from smooth, unimodal symmetry indicated greater sensitivity to data fluctuations, both random and systematic.

UNCERTAINTY IN TRANSFORMED SCORES: MEASUREMENT ERROR, LINKING ERROR, AND SAMPLING ERROR

Error in Scores for Individuals

Whenever an inference is based on a sample, the effect of sampling error on the results must be considered. This error affects the stability of the linking function, so it is referred to here as *linking error*. The linking error describes uncertainty about the score equivalent produced by the linking function and must be included when describing the expected score equivalent for any individual. Moreover, the score for an individual on the original scale also contains uncertainty, described by the measurement error. Thus, when rescaling an individual score, two sources of error must be considered:

1. measurement error
2. linking error

The variance for an individual score that has been rescaled ($var_{individual}$) is estimated by combining the linking error variance ($var_{linking}$) and measurement error variance ($var_{measurement}$):

$$var_{individual}[e_y(x)] = var_{linking}[e_y(x)] + var_{measurement}[e_y(x)]. \quad (35)$$

Linking Error

Linking error for the results obtained in the present study are estimated using Lord's equation (see Kolen and Brennan 1995, p. 227) for estimating the standard error of equipercentile equating using the random group design. The final result of the score transformation is the product of many parameters, each estimated from the data with some sampling error. The final error component in the transformed score is the sum of the errors and their covariances introduced during the estimation of each parameter. The current study assumes a continuous, rather than discrete, score distribution, so the elements in Lord's formula representing approximations of CDF and PDF values from discrete distributions have been replaced by their continuous distribution versions. The revised Equation (36) represents the uncertainty in estimating a score on scale Y from a given score x_i on scale X, $\text{var}_{\text{linkng}}[e_y(x_i)]$:

$$\text{var}_{linking}[\hat{e}_y(x_i)] = \frac{1}{[PDF(y)]^2}\left\{\frac{CDF(x_i)[1 - CDF(x_i)](n_x + n_y)}{n_x n_y} - \frac{[CDF(y) - CDF(x)][CDF(x_i) - CDF(y)]}{n_y[PDF(y)]}\right\}, \quad (36)$$

where PDF(y) is the proportion of examinees estimated to have scale score y, and CDF(x) and CDF(y) are the proportions of examinees estimated to have scale scores at or below x and y, respectively.

The linking error is used to represent the uncertainty of how a score, x, on scale X links to a score, y, on scale Y., given that the sample of examinees used to estimate the transformation function is only randomly equivalent to the full population. The linking error represents how different the results of the linking are expected to be it had been estimated using a different sample of examinees.

Measurement Error

For the assessment methods that the methods explored in this study are expected to be applied, there are three main methods of quantifying error of measurement:

1. Each score is assigned a common measurement error, equal to the standard deviation of normally distributed error specific to the full test form;
2. Each score is assigned a specific measurement error, equal to the standard deviation of a presumed normally distributed[1] error specific to the items each examinee responds to and their proficiency (e.g., FSA); and
3. Each examinee is assigned several scores, which together represent the probability distribution of their proficiency (e.g., PISA).

Measurement error is expressed in terms of the scale of the original score. Consequently, in order to express measurement error of the transformed score, $e_y(x_i)$, the individual measurement variance errors of scale X must first be converted to the scale of scale Y by applying the linking function to the measurement error variance. For the first two scenarios, the variance error of measurement can be rescaled directly using the following formula:

$$\text{var}_{measurement}\left[e_y(x)\right] = \text{var}(x)\left[\frac{(1-PDF(x))PDF(x)}{(1-PDF(e_y x))PDF(e_y x)}\right]^2. \quad (37)$$

Equation 37 describes the multiplication of the variance error of score x on scale X by the ratio of the distribution variance of score x to the distribution variance of the rescaled score, $e_X y$, on scale Y.

For the third scenario, each of the original imputations must be rescaled separately. The set of rescaled scores will represent the posterior density of proficiency for each examinee on the transformed scale. That is,

[1] Error is not always normally distributed, because fixed test forms are likely to provide more information around average ability levels (see Lord, 1983; Warm, 1989).

$$\mathrm{var}_{measurement}\left[e_y\left(pv_{x1}, pv_{x2}, \ldots, pv_{xM}\right)\right] \approx \left(1+\frac{1}{M}\right)\frac{1}{M-1}\sum_{i=1}^{M}\left(e_y pv_{xi} - \overline{e_y pv_x}\right)^2 \quad (38)$$

for the set of M plausible values (adapted from Mislevy, 1991), where $e_y pv_{xi}$ is used to denote the scale *Y* equivalent for *i*th plausible value drawn for an examinee on scale *X*, and $\overline{e_y pv_x}$ is the mean of the five scale *Y* equivalents. Alternately, the same results can be achieved by using the average of the plausible values as a single point estimate for each case. Estimating the measurement variance on the original scale (using Equation 2 with the original plausible values) and converting this variance with Equation 37 will produce approximately the same estimate of converted variance as Equation 38. However, any use of plausible values at the individual level will be inaccurate. Each plausible value represents a random draw from an individual posterior density, and their mean is an approximation to the mean of the posterior density. As this can be estimated directly, it is recommended that any rescaling for interpretation at the individual level should use directly estimated point estimates (e. g., maximum likelihood, weighted likelihood, *expected a posteriori*) and their corresponding parametric measurement errors.

Error in Estimated Sample Statistics

The purpose of this type of rescaling exercising, as previously described, is to produce group level statistics rather than individual estimates. In order to estimate group statistics from rescaled scores, three sources of error must be considered:

1. measurement error
2. linking error
3. sampling error

The goal in this section is to define a method for combining the various sources of error to describing the uncertainty of a statistic estimated from the sample. The general form of the equation for estimating the total variance in a statistic is:

Variance = measurement variance + linking variance + sampling variance (39)

Measurement and Linking Variance

The first two variance components can be estimated simultaneously using Equation 35, which combines measurement and linking variance at the individual level. The group combined variance is estimated by recreating a proficiency density function for each individual based on an available point estimate and its combined errors of measurement and linking. These individual distributions will be summed for the entire group to produce a continuous function describing the distribution of sample scores:

$$P(e_x y) = \frac{1}{n}\sum_{i}^{n} p_i(e_x y), \text{ where} \quad (40)$$

$$p_i(e_x y) = \left[\text{var}_{individual}(e_x y_i)^{\frac{1}{2}}\sqrt{2\pi}\right]^{-1} e^{-\frac{1}{2}\left((e_x y - e_x y_i)/\text{var}_{individual}(e_x y_i)^{\frac{1}{2}}\right)^2}. \quad (41)$$

Although the derivation of all commonly used statistics is outside the scope of this study, two examples of estimation will be presented: the sample mean and the proportion of the sample between two cut-points, along with their respective standard errors.

Combined Error for Means

The mean is defined as the first central moment of the continuous function, $P(e_x y)$, which is analytically defined by Equation 29 and estimated empirically by the summation in Equation 25. The sample score distribution is equal to the sum of the individual score distributions weighted by their individual probability. Individual errors are assumed independent; hence, the covariances between scores are assumed to equal 0. Each individual score has a distinct combined measurement and linking error. Therefore, the combined measurement and linking variance of the mean is equal to the sum of the variances, weighted by the square of their individual probability:

$$\mathrm{var}_{meas.link.}\left(\overline{e_x y}\right)=\frac{1}{n^2}\sum_{i}^{n}\mathrm{var}_{individual}\left(e_x y\right). \quad (42)$$

Combined Error for Proportions

The calculations for error of estimate of proportions are similar. However, each individual contribution to the sample statistic is calculated as the definite integral, $\pi_i(e_x y \mid \mathrm{L1}, \mathrm{L2})$, which describes the probability that the *i*th individual's true score lies between the limits of integration, L1 and L2:

$$\pi_i\left(e_x y|\mathrm{L1},\mathrm{L2}\right)=\int_{\mathrm{L1}}^{\mathrm{L2}} p_i\left(e_x y\right)\partial e_x y. \quad (43)$$

The sample statistic for the proportion of individuals with scores between these cut-points, $\prod(e_x y \mid \mathrm{L1}, \mathrm{L2})$, is estimated as the sum of the n definite integrals, weighted by their probabilities:

$$\Pi\left(e_x y|\mathrm{L1},\mathrm{L2}\right)=\frac{1}{n}\sum_{i}^{n}\pi_i\left(e_x y|\mathrm{L1},\mathrm{L2}\right) \quad (44)$$

Under the assumption of independence of observations, the combined measurement and linking error of the proportion defined in Equation 44 is estimated as the sum of variances of each of the n definite integrals, weighted by the square of their probability.

$$\mathrm{var}\left[\Pi\left(e_x y|l1,l2\right)\right]=\frac{1}{n^2}\sum_{i}^{n}\pi_i\left(e_x y|l1,l2\right)\left[1-\pi_i\left(e_x y|l1,l2\right)\right] \quad (45)$$

Sampling Variance

The sampling variance of the statistic is estimated independently of measurement and linking error. However, estimation of sampling variance typically depends on some knowledge and consideration of the sample design. For the FSA, reporting of provincial results assumes that the students represent a

census, and no sampling error is calculated. For simple random samples, it is expected to be proportional to the sample standard deviation.

For complex samples, typically an appropriate estimation strategy based on Taylor Series approximation or replication is used. For example, sampling variance estimation for PISA analysis uses a balanced repeated replicate (BRR) method (see the *PISA 2000 Technical Report,* 2002). The error variance estimates from each of the above procedures are summed to produce the overall parameter variance. Thus, Equation 39 is operationalized in this situation as:

$$\mathrm{var}_{total}\left(\overline{e_x y}\right) = \mathrm{var}_{meas.link.}\left(\overline{e_x y}\right) + \mathrm{var}_{sampling}\left(\overline{e_x y}\right) . \qquad (46)$$

EVALUATION OF LINKING PROCEDURES

Two criteria are used to evaluate the accuracy of the linking procedure: moment reproduction and the linking error at important reference points on the two scales. The primary goal of the procedure is to rescale scores; therefore the best procedure will be the one that produces rescaled scores that mimic the distributional properties of the target distribution. The criterion of moment reproduction – success in reproducing the empirical moments of the target distribution, $P(y)$, from the rescaled distribution, $P(e_x y)$ – is used to compare the overall effectiveness of each method. The most effective method should produce rescaled score distributions very similar to the original target score distributions. No single method is likely to appear most effective across all indicators. In general, the judgment of comparative effectiveness according to the criterion of moment reproduction follows the same method as for the evaluation of moment preservation of fitted-to-empirical central moments.

Another criterion is suggested by the intended use of the transformed scores – estimation of proportions and other summary statistics at the population level. The estimation of proportions depends on the definition of cut-points; therefore, greater accuracy of the linking procedure around specific cut-points indicates greater utility of the procedure. Kolen and Brennan (1995, chapter 7)

suggest a tolerable limit for the standard error at a cut-point as 0.1 sample standard deviations. For example, if the standard deviation of the scores were 100, a tolerable level of error introduced by the rescaling would be less than 10 at a specific cut-point. For the present study, a standard error of linking of 9 or less (for standard deviations of 92.6 and 91.1) around each of the cut-points for each assessment indicated a reasonably accurate rescaling procedure.

COMPUTER PROGRAMS

All of the required analyses were performed using computing algorithms prepared specifically for this study. The code for each algorithm was written in the *Mathematica* programming language, to be implemented in *Mathematica 4.0* (Wolfram Research, 2000). The complete code, along with detailed explanations of each algorithm, is presented in the Appendix. This code was written and verified with the help of Dr. Steve Hunka, from the Univesity of Alberta, an expert in programming and implementation of Mathematica code (personal communication).

RESULTS

The results of density estimation from each method on each of the test samples are presented in Table 2. For both the FSA and PISA, the method that produced the consistently best fitting models, based on all evaluation criteria, was the fixed mixture model. It was the most accurate in preserving all FSA central moments and consistently ranked first or second in terms of absolute deviation across all eight FSA and PISA central moments and log-likelihood. The kernel methods also had higher log-likelihoods than the 4-parameter Beta and variable mixture methods. This outcome is expected, as the kernel and fixed mixture methods are the most sensitive to observed fluctuations in the data. Although the variable mixture, kernel, and 4-parameter Beta methods tended to be inconsistent in their rankings across all eight central moments, the 4-parameter Beta method

was the most effective in preserving all PISA central moments. The kernel method was consistently ranked third or fourth in moment preservation, and, when it was ranked lowest, its lack of fit in standard deviations were typically an order of magnitude greater than those of the other methods. In general, all methods accurately preserved the central moments, with the fixed mixture model producing the most accurate results for the FSA data, and the 4-parameter Beta model producing the most accurate results for the PISA data.

Table 2

Comparison of Empirical Central Moments to Fitted Central Moments across Estimation Techniques.

FSA	n^*=2659				
	Mean	St. dev.	Skewness	Kurtosis	Log-Likelihood
Empirical Target	509.515	92.556	-0.02095	2.7689	na
Kernel	509.453	95.937	-0.02676	2.8135	-15802.0
Variable mixture	509.468	92.419	-0.00691	2.7934	-15807.8
Fixed mixture	509.518	92.492	-0.01906	2.7782	-15799.9
4-parameter Beta	509.419	92.462	-0.02443	2.9849	-15811.8
PISA	n^*=13,295				
	Mean	St. dev.	Skewness	Kurtosis	Log-Likelihood
Empirical Target	546.875	91.059	-0.25509	3.0962	na
Kernel	546.480	93.257	-0.26236	3.1661	-78752.2
Variable mixture	546.692	90.892	-0.23931	2.9926	-78769.4
Fixed mixture	546.789	90.901	-0.24723	3.0494	-78755.6
4-parameter Beta	546.830	91.027	-0.25595	3.0929	-78772. 4

Note. n* refers to the number of data points used to estimate the density function. Each PISA subject provides 5 data points (n^*_{PISA}=5n).

Figure 1 compares the continuous density estimators from each of the four different methods. All four methods produced smooth densities. However, both the fixed mixture model and the kernel estimator have a noticeable, but smooth, "bump" centred on 700. The smoothest estimator is the 4-parameter Beta.

Figure 1. Comparison of continuous density estimation methods

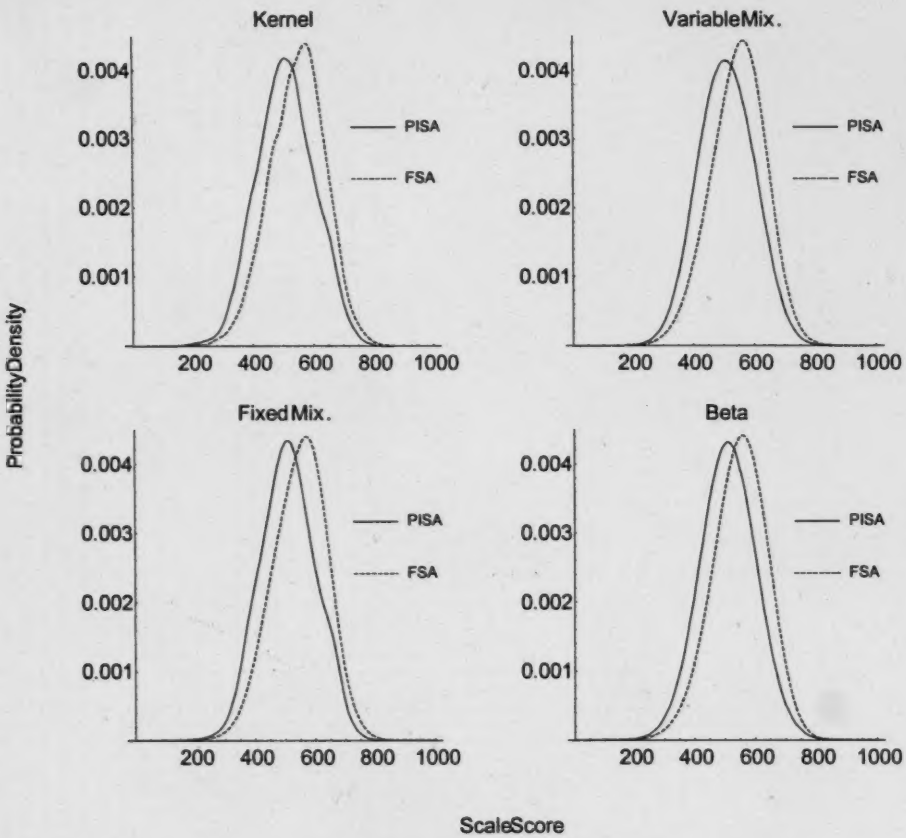


Table 3 summarises the distributional characteristics of the rescaled scores using each density estimation method. All three methods produced comparable results for both FSA-to-PISA linking and vice versa. Similar to the results of the estimation of the initial density estimators, the kernel method was noticeably less

effective in reproducing the standard deviations. No single method was consistently ranked highest in terms of moment reproduction.

Table 3

Distributional Characteristics of Rescaled Scores

FSA-to-PISA				
	Mean	Standard Deviation	Skewness	Kurtosis
PISA Target	546.875	91.059	-0.25509	3.0962
Kernel	547.044	89.150	-0.24440	3.0225
Finite mixture	546.867	91.075	-0.25749	3.0026
Fixed mixture	546.863	91.079	-0.25225	3.0546
4-parameter Beta	546.875	91.053	-0.22974	2.8938
PISA-to-FSA				
	Mean	Standard Deviation	Skewness	Kurtosis
FSA Target	509.515	92.556	-0.02095	2.7689
Kernel	509.473	94.305	-0.02661	2.7884
Finite mixture	509.519	92.494	-0.03101	2.8390
Fixed mixture	509.525	92.527	-0.01941	2.7809
4-parameter Beta	509.814	92.543	-0.02096	3.0052

Figure 2 compares the standard errors of linking for each of the four methods. In general, the methods are equally robust to random error. No particular model was consistently better or worse across the normal range of scores. All four methods produced standard errors less than the tolerable limit of nine across the majority of the score continua. The estimated standard error only exceeds tolerability at around two standard deviations from the mean. In these regions, where observed scores are sparse and required extrapolation of the data, the standard error rapidly approaches infinity. However, these scores are likely to apply to less than 1% of the population, so the large standard errors are not

problematic. Of particular interest are the standard errors of rescaled scores around the cut-points determining performance levels for either assessment. The cut-points, with their corresponding standard errors, are presented in Table 4. Only four of the 28 standard errors across all methods are greater than two-thirds the tolerable limit of nine. The four exceptions are for the PISA Performance Level 1. Because the British Columbia PISA sample had very few students near the bottom of the PISA performance scale, the linking relationship was estimated with more error at the lower end. However, despite being larger, the level of error is still tolerable across all four methods. Therefore, the rescaled scores can be categorised using the original cut-points with acceptable error using any of the four methods.

Figure 2. Comparison of standard error of linking functions for FSA-to-PISA and PISA-to-FSA

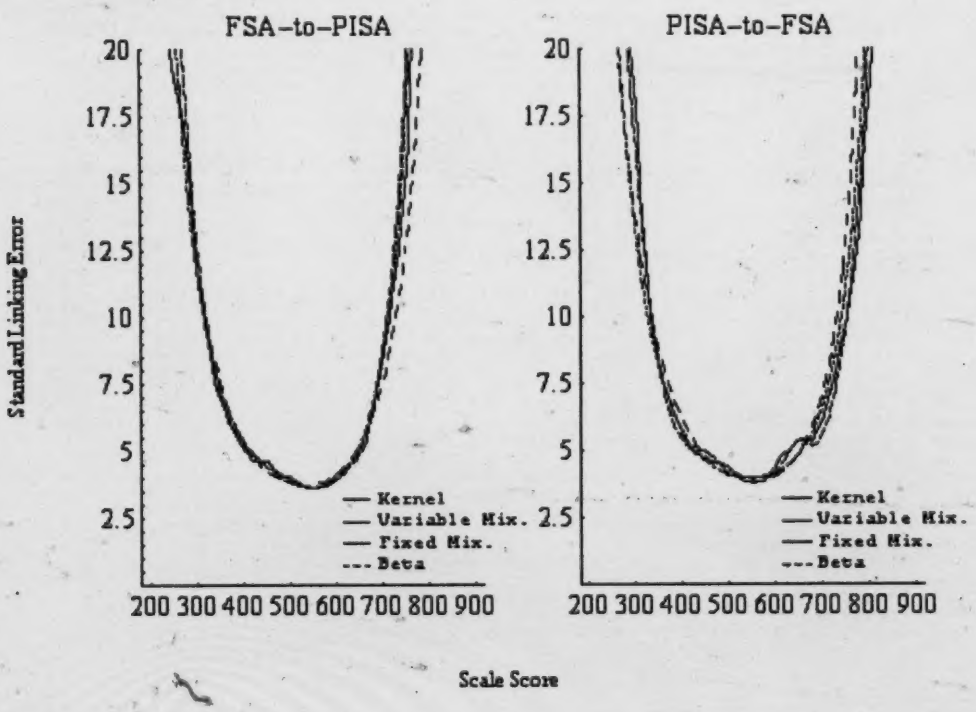


Table 4

Standard Errors of Rescaling Corresponding to Performance Level Cut-points

	Cut-points	Standard Error			
		Kernel	Finite mix.	Fixed mix.	Beta
FSA-to-PISA					
Performance Level 1	335	8.16	8.46	8.49	8.82
Performance Level 2	408	4.95	5.27	5.09	5.21
Performance Level 3	481	4.17	4.06	4.16	3.97
Performance Level 4	553	3.67	3.70	3.68	3.79
Performance Level 5	625	4.42	4.44	4.33	4.59
PISA-to-FSA					
Meeting Expectations	432. 91	5.03	4.94	4.88	5.20
Exceeding Expectations	636. 88	5.15	4.60	5.18	4.61

Alternately, it may be desirable to estimate scale equivalents for the cut-points themselves. That is, rather than rescaling the individual scores in order to categorise against known criteria, it may be more efficient in certain situations to rescale the criteria and categorise the original scores. However, no method currently exists for estimating the uncertainty in estimating a proportion when the cut-point itself is uncertain (while a derivation is possible, the implementation of it may prove computationally expensive). Thus, the rescaled cut-point would be assumed to be a singularity at the rescaled value. Therefore, it is even more important that the error introduced by rescaling is below the criterion for tolerability, because it will be assumed to equal zero. The rescaled values for each cut-point are given in Table 5, with their corresponding standard errors. While the consequences of the use of each cut-point will determine whether the error is

negligible, the values are all below the limit of tolerability, and are approximately one-fifth the size of the original measurement errors at the same cut-points[2].

Table 5

Results of Applying Linking Functions to Transform Cut-points

		Kernel		Variable mixture		Fixed mixture		Beta	
Original cut-points			(s. e.)		(s. e.)		(s. e.)		(s. e.)
PISA-to-FSA									
Level 1	335	368.33	(6.32)	363.13	(6.79)	363.00	(6.51)	366.47	(6.79)
Level 2	408	449.38	(4.69)	446.25	(4.52)	446.89	(4.55)	446.53	(4.38)
Level 3	481	523. 67	(3.81)	524.30	(3.74)	523.17	(3.77)	522.11	(3.77)
Level 4	553	592.87	(4.03)	591.56	(3.91)	593.89	(3.87)	592.31	(4.07)
Level 5	625	652.40	(4.89)	656.29	(5.35)	652.82	(5.10)	659.12	(5.46)
FSA-to-PISA									
Meeting	432. 91	473.78	(4.26)	473.85	(4.14)	473.00	(4.25)	472.82	(4.04)
Exceeding	636. 88	661.43	(5.19)	666.02	(5.78)	661.86	(5.51)	668.71	(5.80)

DISCUSSION

In the absence of an ideal linking design, where both the item administration and sampling are designed with the intent of producing equivalent scales, equipercentile linking is the most appropriate method for linking non-normal or differently shaped score distributions. The results of this study indicate that rescaling of regional and inter-regional assessments is feasible, given the

[2] Although the error of measurement varies across the score continuum, examination of the original scores and their standard errors suggest the range is generally between 20 to 30 points for both assessments around these cut-points.

considerations and assumptions described. The methods developed here for both density estimation and linking introduce minimal error. However, at extremes of the scales where observed scores are sparse, extrapolation of the available data does not produce robust results.

Various methods of continuising discrete observed score distributions were compared in this study. All methods performed well by all criteria, but the kernel method was least effective in terms of moment preservation with these data. Although the mixture models offer the best bias-variance trade-off, there is probably no ideal continuisation procedure that will work best in any situation. For observed score distributions whose only departure from normality is in skewness and kurtosis, the 4-parameter Beta method is likely to be the most accurate, because it can accommodate variations in the third and fourth moments and ignore any fluctuations in higher moments. Results from a study of simulated data sets (Cartwright, 2002) showed that, as violations of normality increased in skewness, kurtosis, and modality, mixture models produced more accurate results, and kernel methods were the most accurate for multimodal data that were not asymptotic towards zero at either the lower or upper extremes of scores. Taken together, the algorithms and programs developed for this study form a battery of reasonably accurate methods that could be applied to any linking situation where the expected score distributions may be non-normal.

The number of numerical operations required for each method of estimation affects the speed of estimation. As the number of operations increases, the computational time required to process them also increases. Estimation of the mixture models requires a significantly greater number of operations than the other methods, a result of the repetitive data passes of each EM cycle and Newton-Raphson iteration. For uncompiled Mathematic code in the Windows 2000 operating system using a Pentium III 1Ghz processor and 256 Mb of RAM, the computation times for estimating linking functions in the current study were: kernel method (195 seconds), variable mixture (47,516 seconds), fixed mixture

(5,467), and 4-parameter Beta (370 seconds). For the estimation of statistics and their combined errors, the computation time is affected by the complexity of the density function. Functions with many terms and higher exponents or roots are the most computationally intensive. Given their complexity, mixture models with a large number of components (greater than 5) and kernel methods require the most time to compute statistics and their combined errors. Given the comparable accuracy of the methods used in the current study, the computational speed and simplicity of the 4-parameter Beta model make it the preferred method for estimating the final linking function, the standard error of linking, and the conversion of measurement errors for the formal linkage between the FSA and the PISA. However, as computer technology improves, estimation time will be relatively brief for all methods, and the choice of method should rely more on statistical criteria.

LIMITATIONS OF THE STUDY

Generalizing the results of a scale linkage across heterogeneous subpopulations, such as males and females, occurs under the assumption that the same linking function would be estimated in both subpopulations. The degree to which a linking function is estimated similarly across heterogeneous subpopulations is known as *invariance*. This study did not fully examine the invariance of the four difference methods of estimation. However, the method chosen for linking FSA and PISA based on the current study, 4-parameter Beta, was evaluated for invariance across male, female, aboriginal, and non-aboriginal sub-populations. The results of this evaluation are displayed in Figures 3 and 4.

Figure 3. Invariance of the FSA-to-PISA linking function

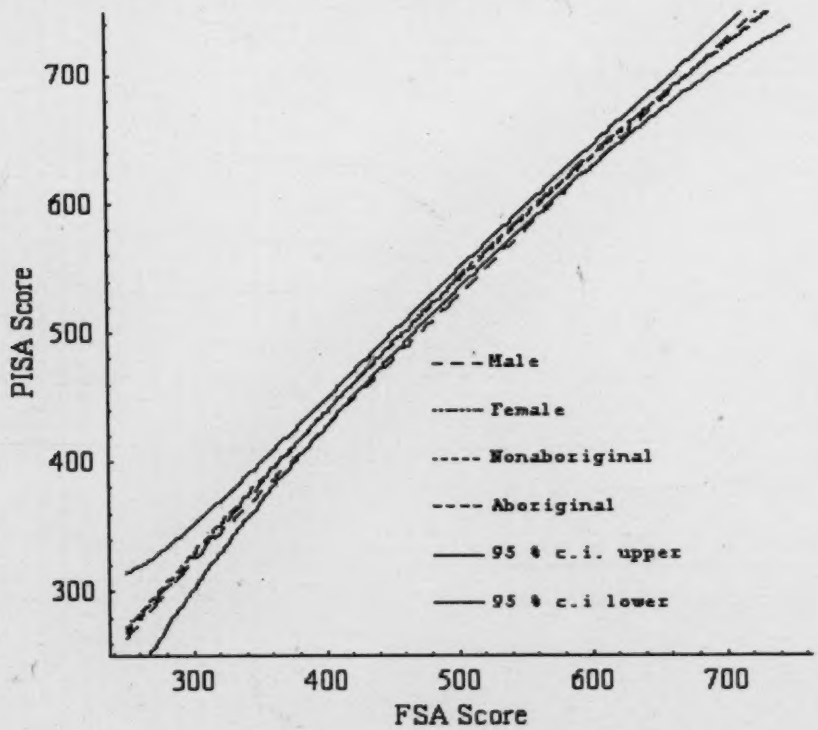


Figure 3 compares the linking functions estimated using each of the four sub-populations to the 95% confidence interval of the linking function estimated using the full sample. This comparison does not account for the larger sampling errors of the sub-population estimates that result from the smaller sample sizes. However, it does illustrate the similarity of the four linking functions across the entire score range. Only the function estimated for males is visually distinguishable from the other three, but at no region of the score continuum are the linkage results for any sub-population significantly different from those for the full sample.

Figure 4. Invariance of the FSA-to-PISA linking function

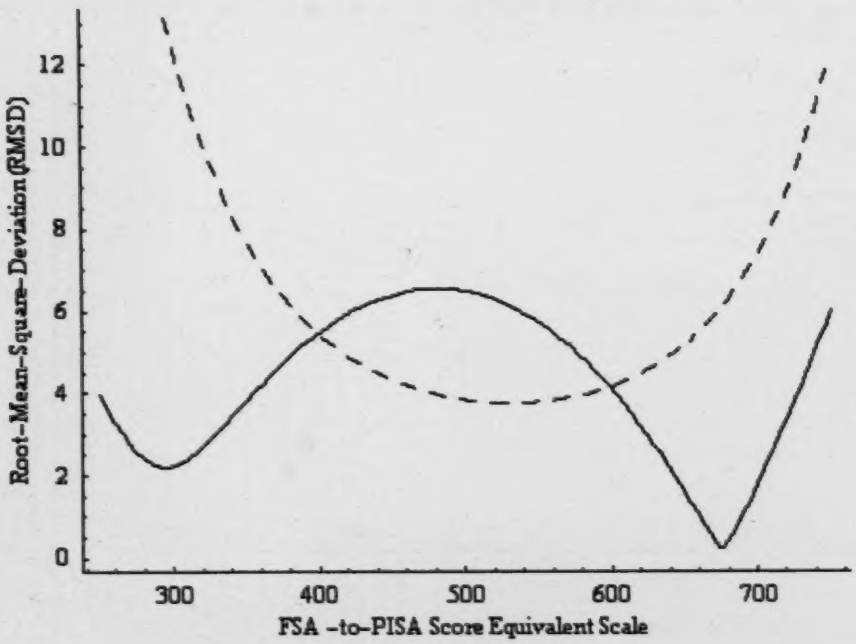


Figure 4 provides a statistical evaluation of invariance. The plot describes the root-mean-square difference (RMSD) between the linking functions computed on each subpopulation and the linking function estimated using the full sample (Dorans and Holland, 2000). This statistic is a function of proficiency and provides a unique value for each score. The dashed line included in the graph for comparison is the standard error of linking. As noted earlier, the standard error has not been corrected to account for the difference in sample sizes, but the RSMD from the 4 heterogeneous sub-populations estimated using smaller samples is of the same magnitude as the full-sample linking error, and smaller across the majority of the scale continuum. This comparison suggests that the systematic variation of the FSA-to-PISA linkage is no greater than would be expected due to chance alone.

In order to maximize invariance and reduce bias, a linking design should include representative sampling from the populations to which the linking function is to be generalized. For example, for the assessments used in this study, the FSA would be administered to an equivalent sample of 15 year-olds as the PISA sample, or PISA would be administered to a representative sample of BC Grade 10 students. Unfortunately, respondent burden and the political separation of assessments make it unlikely that the tests would be administered in a sufficiently short time period and to the same students to allow the application of IRT equating methods. However, the assumptions underlying the methods examined in the current study would still be applicable, and representative sampling would increase the generalizability of any inferences.

Although the similarity of the linking functions in Figure 3 suggests the methods may be applied to substantially smaller sample sizes, this study did not examine the effect of sample size on either the random or bias error of the linking function. The estimation of linking error used for this study (Equation 36) determines a specific relationship between sample size and linking error, but there is no such precise relationship between sample size and the effectiveness of the density estimation methods. However, Cartwright (2002) suggests that the effectiveness of all methods decreases more with small sample sizes when the samples are drawn from multi-modal parent distributions.

FUTURE RESEARCH

Mixture Models

Mixture models, both variable and fixed, appear adequate for modelling a wide variety of distributions, particularly when continuous data are available. However, there are a large number of parameters that require adjustment in order to optimize the estimation procedure for any data set. For both the types of mixture models, parameters include the balance between EM cycles and Newton-Raphson iterations, the number of components, and the assignment of starting

values. Given the number of combinations between the various options, it would be more efficient to determine the optimal values of these parameters without resorting to trial-and-error. Future research with simulated data is needed to develop these optimization solutions.

The algorithms used in this study for mixture model estimation are extremely computationally intensive. A substantial amount of work into mixture model estimation has been conducted in other fields, particularly biometrics and radio telescopy. The adaptation of faster algorithms for the estimation of score distributions would facilitate exploration in this area. Alternate methods may also be used or combined to produce mixture models of the population density, such as cluster analysis and mixed (random effects) model estimation.

Linking Items Using Score-Based Equipercentile Linking Functions

Given the few assumptions required to estimate equipercentile linking functions, the methods used in the current study may also be applicable to the linking of items from different assessments. In order to link assessments at an item level when common examinees are not available, a more complex treatment of the linking function can be used to transform the parameters defining the properties of individual items on one scale into their equivalents on the other scale. In this manner, it will be possible to achieve the goal of IRT equating, which is to rescale each item, even when response data are not available. For location parameters, this will simply be an adjustment the same as that done for examinee scores. Equation 47 demonstrates the relationship between the original difficulty parameter, b_x, and the rescaled parameter, $\hat{b}_y$:

$$\hat{b}_y = e_y(b_x). \tag{47}$$

For discrimination parameters, the relationship is similar to that for the transformation of individual measurement error. The transformed discrimination parameter, $\hat{a}_y$, is found by applying the ratio of distribution variances of the

original scale at the original value to the transformed scale at the transformed value:

$$\hat{a}_y[a_x|b_x] = a_x\left[\frac{(1-PDF(b_x))PDF(b_x)}{(1-PDF(\hat{b}_y))PDF((\hat{b}_y))}\right]^2. \qquad (48)$$

If comparison of the results produced by Equations 47 and 48 are comparable to results produced by simultaneous administration of items, their use may save resources in the maintenance or creation of item banks. Satisfying the assumption of equivalence between samples may be more efficient in certain situations than satisfying the assumptions of equivalent administration conditions and fixed proficiency required for most IRT linkages.

Validity of Inter-regional Linkages

Some directions for further developments in regional and inter-regional assessment linkages are suggested by these results. In particular, more examination is required into the extent of valid interpretations of the linkage function. The current study is limited by the sample bias. While the large sampling fractions ensured small standard errors, and the density estimation methods introduced little estimation bias, the non-random nature of the sample itself may reduce the generalizability of the results. For example, the linkage estimated in the current study may be used to answer questions of the sort, "If a different region had written the FSA, how they have performed?"

This question may be valid for other populations that are similar to the population for which the linkage was estimated, such as other Canadian provinces or countries with similar social and economic climates. However, there is a continuum of validity for certain inferences, and the countries in which PISA was administered are extremely diverse across many social, economic, and cultural indicators. As the systematic differences between the linking population (i.e., 15-year-olds in Grade 10 in British Columbia) and the region to which these results are generalised increase, the validity of interpretations decreases. Currently, there

are no benchmarks or formal methods for evaluating how different two regions can be while still making valid inferences about a linkage that was established in only one of them. As the demand for such comparisons increases, the need for established limits and benchmarks for the validity of such interpretations becomes crucial.

REFERENCES

Adams, R., Wilson, M., & Wang, M. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement, 21*, 1-24.

Adams, R., Wilson, M., & Wu, M. (1997). Multilevel item response models: An approach to errors in variables regression. *Journal of Educational and Behavioural Statistics, 22*, 47-76.

Beaton, A. E., and Gonzalez, E. J. (1993). *Comparing the NAEP trial state assessment results with the IAEP international results.* Paper prepared for the Panel on the Evaluation of the NAEP Trial State Assessment of the National Academy of Education.

Braun, H.I. & Holland, P.W. (1982). Observed-score test equating: A mathematical analysis of some ETS equating procedures. In P. W. Holland and D. B. Rubin (Eds.), *Test Equating* (pp. 9-49). New York: Academic.

British Columbia Ministry of Education. (2000). *Interpreting and communicating British Columbia Foundation Skills Assessment Results.* (Library of Canada publication reference LC1035.8.C32B74 2000). British Columbia: Author.

Bryk, A. S., & Raudenbush, S. W. (1992). *Hierarchical Linear Models.* California:Sage.

Calderone, J., King, L. M., & Horkay, N. (eds). (1997). *The NAEP Guide.* Washington, DC: National Center for Education Statistics.

Cartwright, F. , M., Lalancette, D., Mussio, J, & Xing, D. (in press). Linking provincial student assessments with national and international assessments. (Education, skills and learning Research Paper XX-XXX-XXXXXXXXXX). Statistics Canada: Ottawa.

Cartwright, F. (2002). A comparison of methods for estimating continuous non-normal score distributions. Poster presented at the Measurement for the Social Sciences conference, OISE/UT, Toronto.

Cope, R. T., & Kolen, M. J. (1990). *A Study of Methods for Estimating Distributions of Test Scores.* Iowa: American College Testing.

Crocker, L., & Algina, J. (1986). *Introduction to Classical and Modern Test Theory.* Belmont, CA: Wadsworth Group.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, 39 (Series B),* 1-38.

Dorans, N.J. and Holland, P.W. (2000). Population Invariance and the Equatability of Tests: Basic Theory and the Linear Case. *Journal of Educational Measurement, 37,* 281-306.

Ercikan, K. (1997). Linking statewide tests to the National Assessment of Educational Progress: Accuracy of combining test results across states. *Applied Measurement in Education, 10,* 145-159.

Everitt, B. S., & Hand, D. J. (1981). *Finite Mixture Distributions.* New York: Chapman and Hall Ltd.

Hanson, B. (1990). *An Investigation of Methods for Improving Estimation of Test Score Distributions.* Iowa: American College Testing.

Hanson, B. (1991). *Method of Moments Estimates for Four-parameter Beta Compound Binomial Model and the Calculation of Classification Consistency Indexes.* Iowa: American College Testing.

Härdle, W. (1990). *Smoothing Techniques with Implementation in S.* New York: Springer-Verlag.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning.* New York: Springer.

Johnson, E.G., & Owen, E. (1998). *Linking the National Assessment of Educational Progress (NAEP) and the Third International Mathematics and Science Study (TIMSS): A technical report.* Washington, DC: National Center for Education Statistics.

Johnson, J. L., & Kotz, S. (1970). *Continuous Univariate Distributions 2.* Boston: Houghton Mifflin Company.

Kolen, M. J., & Brennan, R. L. (1995). *Test Equating: Methods and Practices.* New York: Springer.

Organisation for Economic Co-operation and Development (OECD). (2002). *Knowledge and Skills for Life: first results from PISA 2000.* Paris, France: Author.

Linn, R. L. (1993). Linking results of distinct assessments. *Applied Measurement in Education, 6,* 83–102.

Linn, R.L. and Kiplinger, V.L. (1994). Linking statewide tests to the National Assessment of Educational Progress: Stability of results. *Applied Measurement in Education, 8,* 135-156.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* Hillsdale, NJ: Erlbaum.

Lord, F. M. (1983). Unbiased estimators of ability parameters, of their variance, and of their parallel forms reliability. *Psychometrika, 48,* 233-245.

McLaughlin, D. H. (1998). *Study of the linkages of 1996 NAEP and state mathematics assessments in four states.* Washington, DC: National Center for Education Statistics.

Mislevy, R. J. (1991). Randomization-based inference about latent variables from complex samples. *Psychometrika, 56,* 177-196.

Mislevy, R. J. (1992). *Linking educational assessments: Concepts, issues, methods, and prospects.* Princeton, NJ: Policy Information Center, Educational Testing Service.

Morris, C.N. (1982). On the foundations of test equating. In P.W. Holland and D.B. Rubin (Eds.), *Test Equating* (pp. 169-191). New York: Academic.

Muller, P.A., Stage, F.K., & Kinzie, J. (2001). Science achievement growth trajectories: Understanding factors related to gender and racial-ethnic

differences in precollege science achievement. *American Educational Research Journal, 38,* 981-1012.

Muraki, E. (1990). Fitting a polytomous item response models to Likert-type data. *Applied Psychological Measurement, 14,* 59-71.

Muraki, E., & Bock, R.D. (1997). PARSCALE: IRT item analysis and test scoring for rating-scale data [Computer Software]. Chicago, IL: Scientific Software International.

National Council on Educational Standards and Testing (NCEST). (1992). *Raising standards for American education.* Washington, DC: Author.

Pashley, P. J., & Phillips, G. W. (1993). *Toward world-class standards: A research study linking international and national assessments*. Princeton, NJ: Educational Testing Service.

Peterson, N. S., Kolen, M. J., and Hoover, H. D. (1989). Scaling, norming, and equating. In R. L. Linn (Ed.), *Educational Measurement* (3rd Ed., pp. 221-262). New York: Macmillan.

PISA 2000 Technical Report. (2002). R. Adams and M. Wu, Ed. Paris: Organisation for Economic Co-operation and Development.

SAIP 2001 Technical Report. (in press). Council of Ministers of Education, Canada (CMEC). Toronto: CMEC.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* New York: Chapman and Hall Ltd.

Vijverber, W. P. M. (2000). Betit: a family that nests probit and logit. *Discussion Paper No. 222, IZA.* . Bonn, Germany: Institute for Study of Learning (IZA).

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika, 54,* 427-450.

Wolfram Research. (2000). Mathematica (Version 4.0) [Computer programming language]. Champaign, IL: Author.

Yen, W.M. (1983). Tau-equivalence and equipercentile equating. *Psychometrika, 48,* 353-369.

APPENDIX

EQUIPERCENTILE LINKING

Author: Fernando Cartwright
Last Modified: March 2003
Platform: *Mathematica* 4.0
This notebook contains the necessary routines for equipercentile linking using 4 types of continuous density estimation for observed score distributions.

REFERENCES

Everitt, B.S., & Hand, D.J. (1981). *Finite Mixture Distributions*. New York: Chapman and Hall Ltd.
Hanson, B. (1991). *Method of Moments Estimates for Four-parameter Beta Compound Binomial Model and the Calculation of Classification Consistency Indexes*. Iowa: American College Testing.
Hardle, W. (1990). Smoothing Techniques with Implementation in S. New York: Springer-Verlag.
Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: data mining, inference , and prediction.*
Hunka, S. (2001). Fitting Non-linear Regression Models to Data Using Newton's Method for Solving Non-linear Equations. Unpublished *Mathematica* module.
Johnson, J.L., & Kotz, S. (1970). *Continuous Univariate Distributions 2*. Boston: Houghton Mifflin Company.
Ramsay, J.O., & Silverman, B. W. (1997). *Functional Data Analysis*. New York: Springer.
Silverman, B.W. (1986). *Density Esitmation for Statistics and Data Analysis*. New York: Chapman and Hall Ltd.

PACKAGES

```
<<Statistics`DescriptiveStatistics`
<<Graphics`Graphics`
<<Statistics`DataManipulation`
<<Statistics`DataSmoothing`
<<Statistics`NormalDistribution`
<<Statistics`ContinuousDistributions`
<<LinearAlgebra`MatrixManipulation`
<<Statistics`NonlinearFit`
<<NumericalMath`PolynomialFit`
<<Statistics`MultiDescriptiveStatistics`
<<Graphics`Legend`
```

UTILITIES

DATA SUMMARIZING AND WEIGHTING

This function takes a vector of data and assigns weights to each unique value. The *round* value bins the data to the increments indicated. This function can be used to 'weight' unweighted data.

```
weighter[data_, round_]:=Module[{freqdata, wghtdata},
freqdata=Frequencies[Round[data/round]*round];
wghtdata=Transpose[RotateRight[Transpose[freqdata], 1]];
Return[wghtdata];
];
```

weightedfreq[data_] calls for the weighted observed data.It ensures that there are no repeated values in the data by compiling and reweighting the data according to frequency counts.

```
weightedfreq[data_]:=Module[{c, d, e, f}, c=Union[data[[All, 1]]];
d=Map[Apply[Plus, Cases[data, {#, ___}][[All, 2]]]&, c];
f=Transpose[{c, d}];
Return[f];
];
```

unicoord[data_] calls for the (x, y) coordinate data.It ensures that there are no repeated x values in the data by averaging the y-values for repeated x-values.

```
unicoord[data_]:=Module[{c, d, e, f}, c=Union[data[[All, 1]]];
d=Map[Mean[Cases[data, {#, ___}][[All, 2]]]&, c];
f=Transpose[{c, d}];
Return[f];
];
```

alginverse[func_, points_, range_] calls for a monotonic function, a scalar number of points to sample, and the range (a vector of length 2) over which these points should be evenly sampled. The result is a linearly interpolated inverse function.

```
alginverse[func_, points_, range_]:=Module[{n, interv, sample, yvals, coords, invfunc},
n=Part[range, 2]-Part[range, 1];
interv=n/points;
Print["Estimating inverse..."];
sample=Table[j, {j, Part[range, 1], Part[range, 2], interv}];
yvals=func/.x->sample;
coords=Transpose[{yvals, sample}];
invfunc=Interpolation[coords, InterpolationOrder->1][x];
(*Plot[invfunc, {x, Part[range, 1], Part[range, 2]}];*)
Return[invfunc];
];
```

EVALUATION OF MODEL FIT

This module contains several routines used for evaluating the fit of an estimated distribution to both the data and a known parent distribution.

The *wmean[data]* function calls for the weighted data and it produces the weighted estimate of the mean as the 1st central moment.

```
wmean[data_]:=N[(1/(Apply[Plus, data[[All, 2]]]))*Apply[Plus, MapThread[#1*#2&,
Transpose[data]]]];
```

The *wcmoment[mean_, data_, moment_]* function produces central data moments; it calls for the mean of the data, as well as the data and the desired central moment.

```
wcmoment[mean_, sd_, data_, moment_]:=(1/(Apply[Plus, data[[All, 2]]]))*Apply[Plus,
MapThread[((#1-mean)/sd)^moment*#2&, Transpose[data]]];
```

The *fiteval[data_, function_, xmin_, xmax_, function2_, hue_, dashing_]* function compares empirical and analytical central moments, calculates the area misfit between the parent and fitted distributions, calculates the log likelihood of observing the data given the fitted function, and produces figures for both the fitted and parent distributions. It calls for the weighted data, the fitted function, the minimum and maximum of the range in which to evaluate statistics, the parent function, and the hue and dashing specifications for the fitted figure.

```
fiteval[data_, function_, xmin_, xmax_, hue_, dashing_]:=Module[{likely1, empmean,
empstdev, empskew, empkurt, fitmean, fitstdev, fitskew, fitkurt, figure},
likely1=Log[function];
likely1=MapThread[#2*likely1/.x->#1&, Transpose[data]];
likely1=Apply[Plus, likely1];
empmean=wmean[data];
empvar=wcmoment[empmean, 1, data, 2];
empsd=Sqrt[empvar*Apply[Plus, Column[data, 2]]/(Apply[Plus, Column[data, 2]]-1)];
empskew=wcmoment[empmean, empsd, data, 3];
empkurt=wcmoment[empmean, empsd, data, 4];
fitmean=NIntegrate[function*x, {x, xmin, xmax}];
fitvar=NIntegrate[(x-fitmean)^2*function, {x, xmin, xmax}];
fitsd=fitvar^.5;
fitskew=NIntegrate[((x-fitmean)/Sqrt[fitvar])^3*function, {x, xmin, xmax}];
fitkurt=NIntegrate[((x-fitmean)/Sqrt[fitvar])^4*function, {x, xmin, xmax}];
Print[Model = , N[function, 6]];
Print[Plot of fitted model:];
figure=Plot[function, {x, xmin, xmax}, PlotStyle->{Thickness[.005], Hue[hue],
Dashing[dashing]}, AxesOrigin->{xmin, 0}, DisplayFunction->Identity];
bign=Apply[Plus, data[[All, 2]]];
```

```
Print[Log Likelihood = , likely1, (estimated from , Length[data], unique datapoints), and
, bign, sum of weights];
Print[Empirical Central Moments];
Print[Mean= , empmean, St. Dev.= , empsd, Skewness= , empskew, Kurtosis= ,
empkurt];
Print[Fitted Central Moments];
Print[Mean= , fitmean, St. Dev.= , fitsd, Skewness= , fitskew, Kurtosis= , fitkurt];
Return[{figure, empsd, empmean}];
];
```

DENSITY ESTIMATORS

MIXTURE MODELS

All of the functions in this set of modules can be run by *emfinitemix[data_, cycles_, mi_, mincomponents_, maxcomponents_]* or *emfixedmix[data_, cycles_, mi_, components_]* (see below). This algorithm follows the EM algorithm for finite mixtures of univariate normal densities, from Chapter 2 of Everitt & Hand (1981), increasing the number of components is automatic for *emfinitmix*, ranging from *mincomponents* to *maxcomponents* (user-specified). For *emfixedmix*, the number and location of the components is fixed aribtitrarily, and only the weights and variances of the components are estimated (similar to a kernel estimator with kernels evaluated at given nodes). The algorithm values likelihood over parsimony, and therefore chooses the mostly likely model in an absolute sense, rather than using the likelihood ratio to determine the best fit. For both functions, *mi* determines the number of EM cycles to perform before starting Newton-Raphson iterations, and *mi* determines the number of Newton-Raphson iteration to perform before producing final values.

base[mean_, sigma_] produces the base function to be mixed. In this case, it is the Gaussian function. It calls for the mean and standard deviation of a normal distribution to produce the equation form of the distribution.

```
base[mean_, sigma_]:=PDF[NormalDistribution[mean, sigma], x];
```

weight[p_, mean_, sigma_] calls for the mean (mean) and standard deviation (sigma) of a component calculate the posterior probability that a value of x was produced by the component. It is the product of the weighted component at x divided by the value of the mixture (function) at x. The output of this is a function.

```
weight[p_, mean_, sigma_]:=(p*base[mean, sigma]);
```

meani[p_, data_, weight_] calls for the mixing weight (*p*) of a component, the observed weighted data (*data*), and the posterior probability that any value x is produced by the component (*weight*). This produces the updated likelihood estimate of the mean of the component. It is the expected value of x, given that x is drawn from the specified component. The output is a scalar.

*meani[p_, data_, weight_]:=(1/(p*Apply[Plus, data[[All, 2]]]))*Apply[Plus, MapThread[(weight/.x->#1)*#1*#2&, Transpose[data]]];*

sigmai[p_, mean_, data_, weight_] calls for the mixing weight (*p*) of a component, the updated mean (*mean*), the observed weighted data (*data*), and the posterior probability that any value x is produced by the component (*weight*). This produces the updated likelihood estimate of the standard deviation of the component. The output is a scalar.

*sigmai[p_, mean_, data_, weight_]:=Sqrt[(1/(p*Apply[Plus, data[[All, 2]]]))*Apply[Plus, MapThread[(weight*(x-mean)^2/.x->#1)*#2&, Transpose[data]]]];*

startpvector[c_] calls for the number of components to be estimated (*c*) to produce starting values for the vector of mixing weights. The output is a vector of numbers.

startpvector[c_]:=Table[1/c, {i, 1, c}];

startmeanvector[c_, min_, max_] requires the number of components to be estimated (*c*), and the minimum and maximum starting means (min, max) to produce starting values for the vector of means. The output is a vector of numbers.

startmeanvector[c_, min_, max_]:=Take[Table[i, {i, min, max, (max-min)/(c+1)}], {2, -2}];

startsigmavector[c_, data_, means_] requires the number of components to be estimated (c), the original data (data), and the minimum and maximum starting means (min, max) to produce starting values for the vector of standard deviations. The output is a vector of numbers.

startsigmavector[c_, stdev_, means_]:=Table[stdev, {i, 1, c}];

VARIABLE MIXTURE

emvariableemix[data_, cycles_, mi_, mincomponents_, maxcomponents_] This function requires the original weighted data, a specified number of EM cycles, a specified number of Newton-Raphson iterations, and the minimum and maximum number of components to mix. The function will call each of the functions above in order to produce maximum likelihood estimates for variable mixture models.

```
emvariablemix[data_, cycles_, mi_, mincomponents_,
maxcomponents_]:=Module[{components, mean, stdev, bign, likely, likely1, likelihoods,
sypvect, sysigmavect, syfunct, paramsymbols, pd, nsv, fsv, jsv, hess, adj, pvector,
meanvector, sigmavector, current, smallest, step, test, test1, parameters, results, joseph,
plato, laotze},
(*set up lists for recording results for each number of components*)
functions={};
likelihoods={};
```

```
parameters={};
bign=Apply[Plus, data[[All, 2]]];
mean=N[Apply[Plus, MapThread[#1*#2&, Transpose[data]]]/bign];
stdev=Sqrt[Apply[Plus, MapThread[(#1-mean)^2*#2&, Transpose[data]]]/(bign-1)];
Do[
 (*produce arbitrary starting values*)
 pvector=startpvector[components];
   meanvector=startmeanvector[components, mean-2*stdev, mean+2*stdev];
   sigmavector=startsigmavector[components, stdev, meanvector];
 (*create symbolic vectors of the 3 parameters for each component*)
 sypvect=ToExpression[Map[StringJoin[{p, ToString[#]}]&, Table[i, {i, 1,
components}]]];
 symeanvect=ToExpression[Map[StringJoin[{m, ToString[#]}]&, Table[i, {i, 1,
components}]]];
sysigmavect=ToExpression[Map[StringJoin[{s, ToString[#]}]&, Table[i, {i, 1,
components}]]];
 (*define the symbolic model with the current number of parameters*)
 syfunct=Apply[Plus, MapThread[#1*base[#2, #3]&, {sypvect, symeanvect,
sysigmavect}]];
 paramsymbols=Flatten[{symeanvect, sypvect, sysigmavect}];
 likely=Log[syfunct];
 (*define the vector of partial derivatives*)
 pd=Simplify[Map[D[likely, #]&, paramsymbols]];
 (*define the Hessian matrix*)
hess=Simplify[Map[D[pd, #]&, paramsymbols]];
 paramresults=Flatten[{meanvector, pvector, sigmavector}];
 (*starting function for EM cycles*)
 function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
 current=0;
 Do[
   currentparams=paramresults;
   posteriors=MapThread[weight[#1, #2, #3]/function&, {pvector, meanvector,
sigmavector}];
   meanvector=MapThread[meani[#1, data, #2]&, {pvector, posteriors}];
   sigmavector=MapThread[sigmai[#1, #2, data, #3]&, {pvector, meanvector,
posteriors}];
   pvector=MapThread[(posteriors*#2/.x->#1)&, Transpose[data]];
   pvector=Apply[Plus, pvector]/bign;
   paramresults=Flatten[{meanvector, pvector, sigmavector}]/.Underflow[]->0;
   (*Define test value for stopping -- negative or zero variances for any component*)
   test=Min[Take[paramresults/.{Indeterminate->0}, -components]];
   If[test<=0, Break[]];
   function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
   current=cycle;
   , {cycle, 1, cycles}];
 paramresults=currentparams;
 nsv=paramresults;
```

```
 Print[EM cycles for estimating , components, components complete! (after , current,
EM cycles)];
 (*Print[Parameter estimates= , paramresults];*)
 (*iterations*)
 step=0;
 Do[
  If[test<=0, Break[]];
  paramresults=nsv;
   fsv=pd/.MapThread[Rule[#1, #2]&, {paramsymbols, nsv}];
  fsv=MapThread[(fsv/.x->#1)*#2&, Transpose[data]];
  fsv=Apply[Plus, fsv];
  (*evaluate Hessian*)
jsv=hess/.MapThread[Rule[#1, #2]&, {paramsymbols, nsv}];
  jsv=MapThread[(jsv/.x->#1)*#2&, Transpose[data]];
  jsv=Apply[Plus, jsv];
  adj=-1*Inverse[jsv].fsv;
   nsv=nsv+adj;
  test=Min[Take[nsv/.{Underflow[]->0, Indeterminate->0}, -components]];
  If[test<=0, Break[]];
  step=iter;
  , {iter, 1, mi}];
 Print[NR iterations for estimating , components, components complete! (after , step, NR
iterations)];
 (*Print[Parameter estimates= , paramresults];*)
 likely1=likely/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}];
 likely1=MapThread[#2*likely1/.x->#1&, Transpose[data]];
 likely1=Apply[Plus, likely1];
 parameters=Join[parameters, {paramresults}]/.Indeterminate->Null;
 likelihoods=Join[likelihoods, {likely1}];
 function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
 functions=(Join[functions, {function}]);
 If[current&leq;
cycles &&steps&Equal;
0, Break[]];
 , {components, mincomponents, maxcomponents}];
joseph=Sort[Transpose[{likelihoods, functions}]/.Indeterminate->-999999999];
plato=Sort[Transpose[{likelihoods, parameters}]/.Indeterminate->-999999999];
Print[TableForm[joseph]];
laotze=Part[Part[joseph, -1], 2];
(*Print[Part[Part[plato, -1], 2]];*)
likely1=MapThread[#2*Log[laotze]/.x->#1&, Transpose[data]];
likely1=Apply[Plus, likely1];
Print[likely1];
Return[laotze];];
```

FIXED MIXTURE

emfixedmix[data_, cycles_, mi_, components_] This function calls for the original weighted data, a specified number of EM cycles, a specified number of NR iterations, and the number of components to mix.The module will call each of the functions above in order to produce maximum likelihood estimates for mixture model weight and variance parameters.It evenly spreads the components between-2 and 2 standard deviations from the sample mean and estimates optimal variances and mixing weights. Conceptually, the fixed mixture is a convolution of the problem of figuring out where the data came from; instead, it assumes an analyst knows where the data came from and focuses on getting more information about these distributions. Although means are not estimated, a large number of components are required to reduce the bias caused by fixing component locations. This is the most computationally intensive method of density estimation, due to the large number of parameters to estimate.

```
emfixedmix[data_, cycles_, mi_, components_]:=Module[{pvector, mean, stdev,
meanvector, sigmavector, function, allparams, posteriors, paramresults, sypvect,
symeanvect, sysigmavect, syfunct, paramsymbols},
(*produce arbitrary starting values*)
pvector=startpvector[components];
bign=Apply[Plus, data[[All, 2]]];
mean=N[Apply[Plus, MapThread[#1*#2&, Transpose[data]]]/bign];
stdev=Sqrt[Apply[Plus, MapThread[(#1-mean)^2*#2&, Transpose[data]]]/(bign-1)];
meanvector=startmeanvector[components, mean-2*stdev, mean+2*stdev];
 sigmavector=startsigmavector[components, stdev, meanvector];
(*create symbolic vectors of the 3 parameters for each component*)
sypvect=ToExpression[Map[StringJoin[{p, ToString[#]}]&, Table[i, {i, 1,
components}]]];
symeanvect=ToExpression[Map[StringJoin[{m, ToString[#]}]&, Table[i, {i, 1,
components}]]];
sysigmavect=ToExpression[Map[StringJoin[{s, ToString[#]}]&, Table[i, {i, 1,
components}]]];
(*define the symbolic model with the current number of parameters*)
syfunct=Apply[Plus, MapThread[#1*base[#2, #3]&, {sypvect, symeanvect,
sysigmavect}]];
syfunct=syfunct/.MapThread[Rule[#1, #2]&, {symeanvect, meanvector}];
paramsymbols=Flatten[{sypvect, sysigmavect}];
likely=Log[syfunct];
(*define the vector of partial derivatives*)
pd=Map[D[likely, #]&, paramsymbols];
(*define the Hessian matrix*)
   hess=Map[D[pd, #]&, paramsymbols];
(*define the test value for stopping -- the smallest estimated standard deviation after the
nth step*)
paramresults=Flatten[{pvector, sigmavector}];
(*starting function for EM cycles*)
function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
Do[
```

```
 currentparams=paramresults;
 posteriors=MapThread[weight[#1, #2, #3]/function&, {pvector, meanvector,
sigmavector}];
 sigmavector=MapThread[sigmai[#1, #2, data, #3]&, {pvector, meanvector,
posteriors}];
 pvector=MapThread[(posteriors*#2/.x->#1)&, Transpose[data]];
 pvector=Apply[Plus, pvector]/bign;
 paramresults=Flatten[{pvector, sigmavector}]/.Underflow[]->0;
 test=Min[Take[paramresults/.{Indeterminate->0}, -2*components]];
 If[test<=0, Break[]];
 function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
 , {cycle, 1, cycles}];
nsv=paramresults;
Do[
 If[test<=0, Break[]];
 paramresults=nsv;
 fsv=pd/.MapThread[Rule[#1, #2]&, {paramsymbols, nsv}];
 fsv=MapThread[(fsv/.x->#1)*#2&, Transpose[data]];
 fsv=Apply[Plus, fsv];
 (*evaluate Hessian*)
  jsv=hess/.MapThread[Rule[#1, #2]&, {paramsymbols, nsv}];
 jsv=MapThread[(jsv/.x->#1)*#2&, Transpose[data]];
 jsv=Apply[Plus, jsv];
 adj=-1*Inverse[jsv].fsv;
  nsv=nsv+adj;
 test=Min[Take[nsv/.{Underflow[]->0, Indeterminate->0}, -components]];
 If[test<=0, Break[]];
 , {iter, 1, mi}];
function=N[syfunct/.MapThread[Rule[#1, #2]&, {paramsymbols, paramresults}]];
Return[function];
];
```

KERNEL ESTIMATOR MODULES

The *allkernels[rawdata_, nodes_, n_, min_, max_]* (described below) combines all of the functions in this module. This algorithm replaces each observed datum with a kernel function, effectively transforming each datum from a singularity into a distribution. The sum of the kernels is sampled at a specified number of nodes and interpolated to create a density function. It is standardised (transformed into a probability density function) by dividing by its definite integral from min to max. The variable kernel method was described in part in Silverman, 1986 (pages 21-22), and the optimal bandwidth is based on the "Better Rule of Thumb" in Hardle (1990). The bandwidth varies according to the nth root of the average distance to the nearest neighbours.

kernelfunc4[data_, nodes_, h_, n_, numnodes_] calls for the weighted data, the nodes at which the kernel functions are sampled, a smoothing parameter or "bandwidth" (equivalent to the standard deviation used for the kernel function), a parameter for

adjusting for nearest-neighbour proximity, and the number of nodes. This function performs the calculation to produce the unstandardized kernel estimator. It performs one pass through the data, compiling the y-ordinates at all nodes for each datum at a time. The final output is the matrix of sample points and corresponding y-ordinates.

```
kernelfunc4[data_, nodes_, h_, n_, numnodes_]:=Module[{allquads, values, freqs,
quads, weights, offsetdata, offsetdata1, offsetdata2, comf, allq2},
  values=data[[All, 1]];
  freqs=data[[All, 2]];
  offsetdata=Abs[Drop[values-RotateLeft[values], -1]];
  offsetdata1=Join[Take[offsetdata, 1], offsetdata];
  offsetdata2=Join[offsetdata, Take[offsetdata, -1]];
  offsetdata=Mean[{offsetdata1, offsetdata2}];
  allquads=Table[0., {c, 1, numnodes}];
  lnd=Length[data];
  h2=h;
n2=n;
numnodes2=numnodes;
(*set up a compile of the Do loops; call the function comf; it returns the variable allq0*)
    comf=Compile[{{nod, _Real, 1}, {valu, _Real, 1}, {offset, _Real, 1}, {h1, _Real}, {n1,
_Integer}, {frq, _Integer, 1}, {numnod, _Integer}, {allq, _Real, 1}}, Module[{allq0, kern,
i, j}, allq0=allq;
  Do[(*over data points*)
Do[(*over # of nodes*) kern=E^-(((nod[[j]]-Part[valu, i])/(h1*Part[offset,
i]^(1/n1)))^2/2)*Part[frq, i];
  allq0[[j]]=allq0[[j]]+kern;
 , {j, 1, numnod}];
, {i, 1, Length[valu]}];
allq0]];
  (*now execute the compiled function*)
  allq2=comf[nodes, values, offsetdata, h2, n2, freqs, numnodes2, allquads];
 Return[weights];
]
```

allkernels[rawdata_, numnodes_, n_, min_, max_] calls for the original data, number of nodes to sample, the parameter for nearest-neighbour adjustment, and the range over which to evaluate the kernel estimator. The function first estimates mean and standard deviation of the data in order to calculate the optimal bandwidth. A vector of evenly spaced sample points is then created and then the *kernelfunc4* function is run. The results are interpolated and integrated to produce the final estimate.

```
allkernels[data_, numnodes_, n_, min_, max_]:=Module[{total, mean, var, stdev, h, a, b,
t, delta, d, g, i, j, likely1},
total=Apply[Plus, Column[data, 2]];
mean=wmean[data];
var=wcmoment[mean, 1, data, 2];
stdev=Sqrt[var*total/(total-1)];
h=(4./3)^(1/5) *stdev*(Apply[Plus, data[[All, 2]]]^(-1/5));
```

```
a=Min[{min, Min[data]}]-3*h;
b=Max[{max, Max[data]}]+3*h;
(*alternately, the values for a and b can be assigned using the range limits -- this option
has been suppressed here*)
(*a=min;*)
(*b=max;*)
delta=(b-a)/numnodes;
t=Table[a+(k+.5)*delta, {k, 0, numnodes-1}];
d=kernelfunc4[data, t, h, n, numnodes];
g=Interpolation[d, InterpolationOrder->1][x];
i=NIntegrate[g, {x, a, b}];
j=g/i;
likely1=MapThread[#2*Log[j]/.x->#1&, Transpose[data]];
likely1=Apply[Plus, likely1];
Print[likely1];
Return[j];
];
```

BETA 4 ESTIMATION

The *betamodel[data_,stans_]* (see below) implements all the functions in this module for density estimation. This algorithm follows the method of moments algorithm for estimating 3 parameters of the 4 parameter Beta distribution, when the 4th parameter is specified (Hanson, 1991). It specifies the lower limit parameter and produces the other three parameters accordingly, using the non-central moments of the sample data. The lower limit is chosen to make the higher moments of the fitted distribution match the higher moments of the sample (as per Hanson, pg. 7-8). The lower limit is chosen as the smallest value that will produce real roots for the alpha and beta parameters of the distribution (given machine precision). It was observed that smaller values of the lower limit will reduce the squared difference between higher moments of the sample and fitted distributions. The algorithm looks for the smallest lower limit along an inverse quadratic path, so that, as the lower limit approaches the mean of the sample, the incremental changes in the test-values become smaller and smaller. If the search cannot find the smallest lower limit, it will search from the highest point and attempt to find the upper limit and a complex solution.

beta4[alpha_, beta_, lower_, upper_] calls for the 4 parameters of the distribution. It produces the distribution in equation form.

```
beta4[alpha_, beta_, lower_, upper_]:=Module[{func},
func=N[( (-lower+x)^(alpha-1) (upper-x)^(beta-1))/((upper-lower)^(alpha+beta-1)
Beta[alpha, beta])]/.{Underflow[]->0, Overflow->0};
Return[func];
];
```

betaestimator3[data_, lower_] produces the method-of-moments estimates of the distribution parameters. It calls for the weighted data and the lower parameter. It is a

function for demonstrating the fit of higher moments, given the specification of the lower parameter. It follows the above estimation steps and produces the difference between the analytical kurtosis and empirical kurtosis.

```
betaestimator3[data_, lower_]:=Module[{moment1, moment2, moment3, moment4,
uppernumerator, upperdenominator, upper, tmoment1, tmoment2, tmoment3, tmoment4,
alpha, beta},
moment1=Apply[Plus, MapThread[#1*#2&, Transpose[data]]]/Apply[Plus, data[[All,
2]]];
moment2=Apply[Plus, MapThread[(#1)^2*#2&, Transpose[data]]]/(Apply[Plus,
data[[All, 2]]]);
moment3=Apply[Plus, MapThread[(#1)^3*#2&, Transpose[data]]]/(Apply[Plus,
data[[All, 2]]]);
moment4=Apply[Plus, MapThread[(#1)^4*#2&, Transpose[data]]]/(Apply[Plus,
data[[All, 2]]]);
uppernumerator=lower*(moment1^2*moment2-
2*moment2^2+moment1*moment3)+moment1*moment2^2-
2*moment1^2*moment3+moment2*moment3;
upperdenominator=lower*(2*moment1^3-
3*moment1*moment2+moment3)+2*moment2^2-moment1^2*moment2-
moment1*moment3;
upper=uppernumerator/upperdenominator;
tdata=MapThread[{(#1-lower)/(upper-lower), #2}&, Transpose[data]];
tmoment1=Apply[Plus, MapThread[#1*#2&, Transpose[tdata]]]/Apply[Plus, tdata[[All,
2]]];
tmoment2=Apply[Plus, MapThread[(#1)^2*#2&, Transpose[tdata]]]/(Apply[Plus,
tdata[[All, 2]]]);
tmoment3=Apply[Plus, MapThread[(#1)^3*#2&, Transpose[tdata]]]/(Apply[Plus,
tdata[[All, 2]]]);
tmoment4=Apply[Plus, MapThread[(#1)^4*#2&, Transpose[tdata]]]/(Apply[Plus,
tdata[[All, 2]]]);
alpha=(tmoment1*(tmoment1-tmoment2))/(tmoment2-tmoment1^2);
beta=((1-tmoment1)*(tmoment1-tmoment2))/(tmoment2-tmoment1^2);
fun=beta4[alpha, beta, lower, upper];
Return[{alpha, beta, lower, upper}];
];
```

betamodel[data_, stans_] calls for the weighted data and a specifed number of standard deviations to use as a starting point in searching for a computationally feasible lower limit. It stops and returns the distribution estimate as soon as a Real solution is found. Error messages are printed otherwise.

```
betamodel[data_, stans_]:=Module[{sds, betaparams, model, cmodel, final, likely1},
sds=-stans;
mean=Apply[Plus, MapThread[#1*#2&, Transpose[data]]]/Apply[Plus, data[[All, 2]]];
stdev=Sqrt[Apply[Plus, MapThread[(#1)^2*#2&, Transpose[data]]]/(Apply[Plus,
data[[All, 2]]]-1)];
Do[
```

```
 betaparams=betaestimator3[data, mean+(sds*stdev)];
 model=beta4[Part[betaparams, 1], Part[betaparams, 2], Part[betaparams, 3],
Part[betaparams, 4]];
 If[MatchQ[model/.x->0, _Real], Break[]];
 sds=N[-stans*iter^-2.5];
, {iter, 1, 100}];
If[MatchQ[model/.x->0, _Real], final=model, Print[Whoops! No method of moments
solution here - trying another solution...]];
If[MatchQ[model/.x->0, _Complex], sds=6000;
 Do[
  betaparams=betaestimator3[data, mean+(sds*stdev)];
  cmodel=beta4[Part[betaparams, 1], Part[betaparams, 2], Part[betaparams, 3],
Part[betaparams, 4]];
  If[MatchQ[cmodel/.x->0, _Real], Break[]];
  sds=N[stans*iter^2.5];
, {iter, 1, 100}];
 final=cmodel, Null];
likely1=MapThread[#2*Log[final]/.x->#1&, Transpose[data]];
likely1=Apply[Plus, likely1];
 Print[likely1];
Return[final];
];
```

EQUATING ALGORITHMS

Equating functions: This set of modules equates the tests, calculates standard errors, and produces pretty graphs for each of the above smoothing methods. Step 1) estimate the distribution Step 2) Integrate the distribution, Step 3) sample the CDF, oversampling at the extremes, to approximate the inverse function Step 4) sample the inverse functions at equal percentiles, oversampling at the extremes, creating a vector of equipercentile equivalents, Step 5) Interpolate the matched values to get an approximate equating function, Step 6) find equated scores, Step 7) Evaluate the process. Evaluation steps include comparing the Form B distribution to the original Form A distribution and the transformed Form A distribution; comparing central moments from the original, target, and transformed distributions; a plot of the standard error of equating across the range of observed scores, and the root mean variance error of equating across the range of observed scores. There are also two utilities that tabulate and compare correlations. The output of each equating function is of the form: (equating function, inverse equating function, error function, inverse error function, original distribution, target distribution)

UTILITIES

distprop[data1_, function1_, data2_, function2_, plotmin_, plotmax_] calls for the first raw data, the first estimated function, the second raw data, the second estiamted function, and minimum and maximum values for plotting results. It examines distribution properties and fit of the smoothed distributions

```
distprop[data1_, function1_, data2_, function2_, plotmin_,
plotmax_]:=Module[{original, target, sda, meana, sdb, meanb, originalplot, targetplot},
Print[Distribution properties...];
Print[Form A: ];
original=fiteval[data1, function1, plotmin, plotmax, .9, {0, 0}];
originalplot=Part[original, 1];
sda=Part[original, 2];
meana=Part[original, 3];
Print[Form B: ];
target=fiteval[data2, function2, plotmin, plotmax, .65, {.025, 0.025}];
targetplot=Part[target, 1];
sdb=Part[target, 2];
meanb=Part[target, 3];
Show[originalplot, targetplot, DisplayFunction->$DisplayFunction];
Return[{sda, meana, sdb, meanb, originalplot, targetplot}];
];
```

equating[data1_, aperc_, meana_, sda_, bperc_, meanb_, sdb_, min_, max_] calls for the orginal data, the CDF for the original scale, the mean and sd of the original data, the CDF for the target scale, mena and sd of the target data, and minimum and maximum for the score range. This function equates the scores from one test into equipercentile equivalents on the other: it returns { equating function1, equating function2, equated scores1, equated scores2 }

```
equating[data1_, aperc_, meana_, sda_, bperc_, meanb_, sdb_, min_,
max_]:=Module[{equiperc1, equfunc, equated},
Print[Mapping...];
equiperc1=Map[{aperc, bperc}/.x->#&, Table[Erf[i]*1/2+1/2, {i, -2.5, 2.5, 0.0025}]];
Print[Equating function...];
equfunc=Interpolation[equiperc1, InterpolationOrder->1][x];
(*Plot[equfunc, {x, min, max}, PlotRange->{min, max}, PlotStyle->{Thickness[.005],
Hue[.8]}, ImageSize->350];*)
Print[Processing equated scores...];
equated=MapThread[{equfunc/.x->#1, #2}&, Transpose[data1]];
Return[{equated, equfunc}];
];
```

equateeval1[name_, equfunc_, data1_, equated_, data2_, a02_, b02_, cdfa_, cdfb_, plotmin_, plotmax_] calls for a string in quotations that names the procedure, the equating function, original data, equated scores, target scores, original PDF, target PDF, original CDF, target CDF, and minimum and maximum for plotting the results of the equating procedure

```
equateeval1[name_, equfunc_, data1_, equated_, data2_, a02_, b02_, cdfa_, cdfb_,
plotmin_, plotmax_]:=Module[{equatedprop, equatedplot, sdequated, meanequated,
minlength, misfit, error, meanerror},
Print[name];
Print[DISTRIBUTION OF PREDICTED FORM B SCORES];
```

```
meanequated=wmean[equated];
empvar=wcmoment[meanequated, 1, equated, 2];
sdequated=Sqrt[empvar*Apply[Plus, Column[equated, 2]]/(Apply[Plus,
Column[equated, 2]]-1)];
empskew=wcmoment[meanequated, sdequated, equated, 3];
empkurt=wcmoment[meanequated, sdequated, equated, 4];
Print[Empirical Central Moments];
Print[Mean= , meanequated, St. Dev.= , sdequated, Skewness= , empskew, Kurtosis= ,
empkurt];
minlength=Min[{Length[data1], Length[data2]}];
error=(b02/.x->(equfunc))^-2*((cdfa*(1-
cdfa)*(minlength+minlength))/(minlength*minlength)-(((cdfb/.x->(equfunc))-
cdfa)*(cdfa-(cdfb/.x->(equfunc)))/(minlength*b02/.x->(equfunc))));
Plot[Sqrt[error], {x, meanequated-1.96*sdequated, meanequated+1.96*sdequated},
PlotRange->{0, sdequated/5}];
wfunc=PDF[NormalDistribution[meanequated, sdequated], x];
meanerror=(NIntegrate[error*wfunc, {x, meanequated-1.96*sdequated,
meanequated+1.96*sdequated}])^.5;
Print[Root Mean error    , meanerror];
Return[error];
];
```

VARIABLE MIXTURE

```
equatervariablemix[dataa_, round1_, datab_, round2_, cycles_, mi_, mincomponents_,
maxcomponents_]:=Module[{data1, data2, a02, b02, cdfa, cdfb, sda, sdb, meana,
meanb, scorevectora, scorevectorb, aperc, bperc, equiperc1, equateres,
equfunc1,equfunc2 equated, originalplot, targetplot, error1,error2},
(*get parameters for showing plots*)
data1=weighter[dataa, round1];
data1=weightedfreq[data1];
data2=weighter[datab, round2];
data2=weightedfreq[data2];
plotmin=Min[Join[dataa, datab]];
plotmax=Max[Join[dataa, datab]];
(*get continuous distributions*)
Print[Estimating continuous distributions...];
a02=emvariablemix[data1, cycles, mi, mincomponents, maxcomponents];
b02=emvariablemix[data2, cycles, mi, mincomponents, maxcomponents];
(*evaluate fitted models*)
dists=distprop[data1, a02, data2, b02, plotmin, plotmax];
sda=Part[dists, 1];
meana=Part[dists, 2];
sdb=Part[dists, 3];
meanb=Part[dists, 4];
originalplot=Part[dists, 5];
targetplot=Part[dists, 6];
```

```
(*estimate inverse functions*)
Print[Integrals...];
cdfa=Integrate[a02, x]+.5;
cdfb=Integrate[b02, x]+.5;
Print[Score vectors...];
scorevectora=Table[(Erf[i*.75] 5*sda)+meana, {i, -2.5, 2.5, .002}];
scorevectorb=Table[(Erf[i*.75] 5*sdb)+meanb, {i, -2.5, 2.5, .002}];
Print[Distribution sampling...];
aperc=Map[{cdfa/.x->#, #}&, scorevectora];
bperc=Map[{cdfb/.x->#, #}&, scorevectorb];
aperc=unicoord[aperc];
bperc=unicoord[bperc];
aperc=Interpolation[aperc, InterpolationOrder->1][x];
bperc=Interpolation[bperc, InterpolationOrder->1][x];
(*equate scores*)
equateres=equating[data1, aperc, meana, sda, bperc, meanb, sdb, plotmin, plotmax];
equated=Part[equateres, 1];
equfunc1=Part[equateres, 2];
equfunc2=alginverse[equfunc1, 5000, {plotmin, plotmax}];
 equated2=equfunc2/.x->data2[[All, 1]];
 equated2=Transpose[{equated2, data2[[All, 2]]}];
 (*evaluate the fit of the equated scores*)
 error1=equateeval1[A-to-B, equfunc1, data1, equated, data2, a02, b02, cdfa, cdfb,
plotmin, plotmax];
 error2=equateeval1[B-to-A, equfunc2, data2, equated2, data1, b02, a02, cdfb, cdfa,
plotmin, plotmax];
 (*end*)
 Return[{equfunc1, equfunc2, error, error2, a02, b02}];
];
```

FIXED MIXTURE

```
equaterfixedemix[dataa_, round1_, datab_, round2_, cycles_, mi_,
components_]:=Module[{data1, data2, a02, b02, cdfa, cdfb, sda, sdb, meana, meanb,
scorevectora, scorevectorb, aperc, bperc, equiperc1, equateres, equfunc1,equfunc2
equated, originalplot, targetplot, error1,error2},
(*get parameters for showing plots*)
data1=weighter[dataa, round1];
data1=weightedfreq[data1];
 data2=weighter[datab, round2];
 data2=weightedfreq[data2];
plotmin=Min[Join[dataa, datab]];
plotmax=Max[Join[dataa, datab]];
rangemin=0;
 rangemax=.005;
 (*get continuous distributions*)
 Print[Estimating continuous distributions...];
```

```
a02=emfixedmix[data1, cycles, mi, components];
b02=emfixedmix[data2, cycles, mi, components];
(*evaluate fitted models*)
dists=distprop[data1, a02, data2, b02, plotmin, plotmax];
sda=Part[dists, 1];
meana=Part[dists, 2];
 sdb=Part[dists, 3];
 meanb=Part[dists, 4];
originalplot=Part[dists, 5];
targetplot=Part[dists, 6];
(*estimate inverse functions*)
Print[Integrals...];
cdfa=Integrate[a02, x]+.5;
cdfb=Integrate[b02, x]+.5;
Print[Score vectors...];
scorevectora=Table[(Erf[i*.75] 5*sda)+meana, {i, -2.5, 2.5, .002}];
scorevectorb=Table[(Erf[i*.75] 5*sdb)+meanb, {i, -2.5, 2.5, .002}];
Print[Distribution sampling...];
aperc=Map[{cdfa/.x->#, #}&, scorevectora];
bperc=Map[{cdfb/.x->#, #}&, scorevectorb];
aperc=unicoord[aperc];
bperc=unicoord[bperc];
 aperc=Interpolation[aperc, InterpolationOrder->1][x];
 bperc=Interpolation[bperc, InterpolationOrder->1][x];
(*equate scores*)
equateres=equating[data1, aperc, meana, sda, bperc, meanb, sdb, plotmin, plotmax];
equated=Part[equateres, 1];
equfunc1=Part[equateres, 2];
equfunc2=alginverse[equfunc1, 5000, {plotmin, plotmax}];
 equated2=equfunc2/.x->data2[[All, 1]];
 equated2=Transpose[{equated2, data2[[All, 2]]}];
 (*evaluate the fit of the equated scores*)
 error1=equateeval1[A-to-B, equfunc1, data1, equated, data2, a02, b02, cdfa, cdfb,
plotmin, plotmax];
 error2=equateeval1[B-to-A, equfunc2, data2, equated2, data1, b02, a02, cdfb, cdfa,
plotmin, plotmax];
 (*end*)
 Return[{equfunc1, equfunc2, error, error2, a02, b02}];
];
```

KERNEL

```
equaterkernel[dataa_, round1_, datab_, round2_, nodes_, n_, min_,
max_]:=Module[{data1, data2, a02, b02, cdfa, cdfb, sda, sdb, meana, meanb,
scorevectora, scorevectorb, aperc, bperc, equiperc1, equateres, equfunc1,equfunc2
equated, originalplot, targetplot, error1,error2},
data1=weighter[dataa, round1];
```

```
data1=weightedfreq[data1];
data2=weighter[datab, round2];
data2=weightedfreq[data2];
(*get parameters for showing plots*)
plotmin=Min[Join[dataa, datab]];
plotmax=Max[Join[dataa, datab]];
(*get continuous distributions*)
Print[Estimating continuous distributions...];
a02=allkernels[data1, nodes, n, min, max];
b02=allkernels[data2, nodes, n, min, max];
(*evaluate fitted models*)
dists=distprop[data1, a02, data2, b02, plotmin, plotmax];
sda=Part[dists, 1];
meana=Part[dists, 2];
sdb=Part[dists, 3];
meanb=Part[dists, 4];
originalplot=Part[dists, 5];
targetplot=Part[dists, 6];
(*estimate inverse functions*)
Print[Integrals...];
cdfa=Integrate[a02, x];
cdfb=Integrate[b02, x];
Print[Score vectors...];
scorevectora=Table[(Erf[i*.75] 5*sda)+meana, {i, -2.5, 2.5, .002}];
scorevectorb=Table[(Erf[i*.75] 5*sdb)+meanb, {i, -2.5, 2.5, .002}];
Print[Distribution sampling...];
aperc=Map[{cdfa/.x->#, #}&, scorevectora];
bperc=Map[{cdfb/.x->#, #}&, scorevectorb];
aperc=unicoord[aperc];
bperc=unicoord[bperc];
aperc=Interpolation[aperc, InterpolationOrder->1][x];
bperc=Interpolation[bperc, InterpolationOrder->1][x];
(*equate scores*)
equateres=equating[data1, aperc, meana, sda, bperc, meanb, sdb, plotmin, plotmax];
equated=Part[equateres, 1];
equfunc1=Part[equateres, 2];
equfunc2=alginverse[equfunc1, 5000, {plotmin, plotmax}];
equated2=equfunc2/.x->data2[[All, 1]];
equated2=Transpose[{equated2, data2[[All, 2]]}];
(*evaluate the fit of the equated scores*)
error1=equateeval1[A-to-B, equfunc1, data1, equated, data2, a02, b02, cdfa, cdfb,
plotmin, plotmax];
error2=equateeval1[B-to-A, equfunc2, data2, equated2, data1, b02, a02, cdfb, cdfa,
plotmin, plotmax];
(*end*)
Return[{equfunc1, equfunc2, error, error2, a02, b02}];
];
```

BETA

```
equaterbeta[dataa_, round1_, datab_, round2_, stans_]:=Module[{data1, data2, a02,
b02, cdfa, cdfb, sda, sdb, meana, meanb, scorevectora, scorevectorb, aperc, bperc,
equiperc1, equateres, equfunc1,equfunc2 equated, originalplot, targetplot,
error1,error2},
(*get parameters for showing plots*)
data1=weighter[dataa, round1];
data1=weightedfreq[data1];
 data2=weighter[datab, round2];
 data2=weightedfreq[data2];
plotmin=Min[Join[data1, data2]];
plotmax=Max[Join[data1, data2]];
 rangemin=0;
 rangemax=.005;
(*get continuous distributions*)
Print[Estimating continuous distributions...];
a02=betamodel[data1, stans];
b02=betamodel[data2, stans];
(*evaluate fitted models*)
dists=distprop[data1, a02, data2, b02, plotmin, plotmax];
sda=Part[dists, 1];
meana=Part[dists, 2];
 sdb=Part[dists, 3];
 meanb=Part[dists, 4];
originalplot=Part[dists, 5];
targetplot=Part[dists, 6];
(*estimate inverse functions*)
scorevectora=Table[(Erf[i*.75] 5*sda)+meana, {i, -2.5, 2.5, .002}];
scorevectorb=Table[(Erf[i*.75] 5*sdb)+meanb, {i, -2.5, 2.5, .002}];
Print[Score vector 1...];
aperc=Map[{NIntegrate[a02, {x, 0, #}], #}&, scorevectora];
aperc=unicoord[aperc];
cdfa=Interpolation[Transpose[RotateRight[Transpose[aperc]]], InterpolationOrder-
>1][x];
Print[Score vector 2...];
bperc=Map[{NIntegrate[b02, {x, 0, #}], #}&, scorevectorb];
bperc=unicoord[bperc];
cdfb=Interpolation[Transpose[RotateRight[Transpose[bperc]]], InterpolationOrder-
>1][x];
aperc=Interpolation[aperc, InterpolationOrder->1][x];
bperc=Interpolation[bperc, InterpolationOrder->1][x];
Print[Distribution sampling...];
(*equate scoers*)
equateres=equating[data1, aperc, meana, sda, bperc, meanb, sdb, plotmin, plotmax];
equated=Part[equateres, 1];
equfunc1=Part[equateres, 2];
```

```
equfunc2=alginverse[equfunc1, 5000, {plotmin, plotmax}];
 equated2=equfunc2/.x->data2[[All, 1]];
 equated2=Transpose[{equated2, data2[[All, 2]]}];
 (*evaluate the fit of the equated scores*)
 error1=equateeval1[A-to-B, equfunc1, data1, equated, data2, a02, b02, cdfa, cdfb,
plotmin, plotmax];
 error2=equateeval1[B-to-A, equfunc2, data2, equated2, data1, b02, a02, cdfb, cdfa,
plotmin, plotmax];
 (*end*)
 Return[{equfunc1, equfunc2, error, error2, a02, b02}];
];
```

IMPLEMENTATION OF LINKING FUNCTIONS (WITH WEIGHTED DATA)

The *cconvert[scores_, results_]* function takes the original scores(s) converts them onto a different scale using the results from the linking procedure. The output includes the following columns (original score, rescaled score, linking error, error conversion factor). The error conversion factor is what the original error must be multiplied by to convert it to the new scale.

```
cconvert[eqresults_, scores_]:=Module[{function, errors, error, dist1, dist2, results},
function=Part[eqresults, 1];
error=Part[eqresults, 2];
dist1=Part[eqresults, 3];
dist2=Part[eqresults, 4];
results=Map[N[{#, function/.x->#, dist1/.x->#}]&, scores];
results=Map[N[Flatten[{Part[#, 1], Part[#, 2], error/.x->Part[#, 2], Part[#, 3], dist2/.x-
>Part[#, 2]}]]&, results];
(* original, rescaled score, rescaling error, y-ordinate of score on scale 1, y-ordinate of
rescaled score on scale 2*)
results=Map[N[Flatten[{Part[#, 1], Part[#, 2], Part[#, 3], (((1-Part[#, 4])*Part[#,
4])/((1-Part[#, 5])*Part[#, 5]))^2}]]&, results];
(* original, rescaled score, rescaling error, ratio of the variances of the scales at the
original score and the rescaled score squared*)
Return[results];
];
```

The *sconvert[scores_, results_]* function takes the original scores(s) and converts them onto a different scale with linking standard errors using the results from the linking procedure.

```
sconvert[eqresults_, scores_]:=Module[{function, errors, error, dist1, dist2, results},
function=Part[eqresults, 1];
error=Part[eqresults, 2];
dist1=Part[eqresults, 3];
dist2=Part[eqresults, 4];
newscores=function/.x->scores;
```

```
errors=error^.5/.x->newscores;
results=Transpose[{newscores, errors}];
results=newscores;
Return[Transpose[{results, errors}]];
];
```

This *pvcompiler[data_, startpv_, endpv_, nweight_]* function takes plausible values data with id codes and replicate weights and coverts the plausible values into point estimates with standard errors. It calls for data of the form (id, pv1, ...pvn, weight1, ..., weightn), as well as the start column of plausible values, the end column of plausible values, and the number of weights.

```
pvcompiler[data_, startpv_, endpv_, nweight_]:=Module[{npv, newdata},
npv=endpv-startpv+1;
newdata=Map[Flatten[{Part[#, 1], Mean[Take[#, {startpv, endpv}]], Variance[Take[#,
{startpv, endpv}]]*(1+1/npv), Take[#, -nweight]}]&, data];
Return[newdata];
];
(*this produces a column of (id, score, error, weight1, weight2..., weightn) vectors*)
```

convertor[eqresults_, data_, nweights_] is a tabulation function. It produces a conversion table for all score values between 100 and 900, at .01 intervals. The output is (id, new score, combined equating error and transformed measurement error, weight). It calls for the conversion function, the error function, the density function of the original scale, the density function of target scale, and the original scores. The original scores are of the form (id, score, error, weight1, weight2, ...).

```
convertor[eqresults_, data_, nweights_]:=Module[{results, function, error, dist1, dist2,
},
function=Part[eqresults, 1];
error=Part[eqresults, 2];
dist1=Part[eqresults, 3];
dist2=Part[eqresults, 4];
results=Map[N[Flatten[{Part[#, 1], function/.x->Part[#, 2], Part[#, 3], dist1/.x-
>Part[#, 2]}]]&, data];
(*id, new score, original measurement error, y-ordinate of original score*)
results=Map[N[Flatten[{Part[#, 1], Part[#, 2], error/.x->Part[#, 2], Part[#, 3], Part[#,
4], dist2/.x->Part[#, 2]}]]&, results];
(*id, new score, linking error, original measurement error, y-ordinate of original score,
y-ordinate of new score*)
results=Map[N[Flatten[{Part[#, 1], Part[#, 2], Part[#, 3]+Part[#, 4]*(((1-Part[#,
5])*Part[#, 5])/((1-Part[#, 6])*Part[#, 6]))^2}]]&, results];
(*id, new score, square of combined linking error and adjusted measurement error*)
results=MapThread[Flatten[{#1, #2}]&, {results, TakeColumns[data, -nweights]}];
Return[results];
];
```

DESCRIPTIVE STATISTICS

The following functions calculate simple descriptive statistics and their standard errors using replication methods or SRS assumptions. They call for the converted scores, the groupid of the group of interest, the number of weights, including the full weight, and, for the proportions, the two cut points defining the proportion. These limits can include out-of-range low or high scores for defining the lowest and highest groups.

descmean[compiled_, groupid_, nweights_] estimates the population mean of the converted scores. The standard error it estimates includes measurement error, linking error, and sampling error.

```
descmean[compiled_, groupid_, nweights_]:=Module[{data, pos, res1, change,
totweights, fullsample, mean1, mesemean, res, rsample, rmean, rres},
data=Cases[compiled, {groupid, ___}];
If[nweights>1, res1={}, res1=0];
change=0;
If[nweights&Equal;
0, pos=1, pos=nweights];
totweights=Apply[Plus, Column[data, -pos]];
fullsample=Transpose[{Column[data, 2], Column[data, -nweights]/totweights}];
mean1=wmean[fullsample];
mesemean=Map[Part[#, 1]*Part[#, 2]^2&, Transpose[{Column[data, 3], Column[data,
-pos]/totweights}]];
mesemean=Apply[Plus, mesemean];
res=mean1;
Do[
 change=change+1;
 totweights=Apply[Plus, Column[data, -nweights+change]];
 rsample=Transpose[{Column[data, 2], Column[data, -nweights+change]/totweights}];
 rmean=wmean[rsample];
 rres=rmean;
 res1=Join[{(rres-res)^2}, res1];
 , {nweights-1}];
res1=Apply[Plus, res1]/80*4;
(*the value of /80*4 here should be changed for the general formula*)
If[nweights&Equal;
1, res1=wcmoment[mean1, 1, compiled, 2]*totweights/(totweights-1)/totweights,
res1=res1];
If[nweights&Equal;
0, res1=0];
res1=Sqrt[mesemean+res1];
 Print[Mean score for Groupid = , groupid, is:];
Print[ mean1, s.e.= , res1];
];
```

descprop[compiled_, groupid_, nweights_, cut1_, cut2_] requires the same input as above, but also two cut points. The function will estimate the proportion of the population

between the cut points and estimate a combined measurement, linking, and sampling error.

```
descprop[compiled_, groupid_, nweights_, cut1_, cut2_]:=Module[{data, pos, res1,
change, totweights, prop1, meseprop, res, rsample, rprop1, rres},
  data=Cases[compiled, {groupid, ___}];
  res1=0;
  change=0;
  If[nweights&Equal;
0, pos=1, pos=nweights];
  totweights=Apply[Plus, Column[data, -pos]];
  prop1=Map[{CDF[NormalDistribution[Part[#, 1], Part[#, 2], cut2]]-
 CDF[NormalDistribution[Part[#, 1], Part[#, 2], cut1]], Part[#, 3]/totweights}&,
Transpose[{Column[data, 2], Column[data, 3], Column[data, -pos]}]];
  meseprop=Apply[Plus, MapThread[#1*(1-#1)*#2^2&, Transpose[prop1]]];
  prop1=Apply[Plus, MapThread[#1*#2&, Transpose[prop1]]];
  res=prop1;
  Do[change=change+1;
totweights=Apply[Plus, Column[data, -nweights+change]];
rsample=Transpose[{Column[data, 2], Column[data, -nweights+change]/totweights}];
rprop1=Map[{CDF[NormalDistribution[Part[#, 1], Part[#, 2], cut2]]-
  CDF[NormalDistribution[Part[#, 1], Part[#, 2], cut1]], Part[#, 3]/totweights}&,
Transpose[{Column[data, 2], Column[data, 3], Column[data, -nweights+change]}]];
rprop1=Apply[Plus, MapThread[#1*#2&, Transpose[rprop1]]];
rres=rprop1;
res1=((rres-res)^2)*1/(20)+res1;
, {nweights-1}];
  (*the factor 1/(20) above should be changed to the general form for different
replication weights*)
  If[nweights&Equal;
1, res1=(1-Length[data]/totweights)*prop1*(1-prop1)/totweights, res1=res1];
  If[nweights&Equal;
0, res1=0, res1=res1];
  res1=Sqrt[meseprop+res1];
  Print[Proportion between , cut1, and , cut2, for Groupid = , groupid, is:];
  Print[ prop1, s.e.= , res1];
  ];
```

The *descs[data_, groupid_, nweights_, cuts_, ncuts_, crit_, lower_, upper_]* function produces the means and proportions within specified ranges, with appropriate standard errors. It calls for the score data (with ids, errors and weights), the group id for the group of interest, the number of weights, the cut values defined performance ranges of interest, the number of cut points, a criterion of interest, a reasonable lower limit of integration, and a reasonable upper limit of integration. There is no output, but the results are printed onscreen.

```
descs[data_, groupid_, nweights_, cuts_, ncuts_, crit_, lower_,
upper_]:=Module[{data2, cuts2},
```

```
data2=Cases[data, {groupid, ___}];
descmean[data2, groupid, nweights];
cuts2=Prepend[cuts, lower];
cuts2=Append[cuts2, upper];
Do[descprop[data2, groupid, nweights, Part[cuts2, i], Part[cuts2, i+1]];
  , {i, 1, ncuts+1}];
descprop[data2, groupid, nweights, crit, upper];
];
```